Relation Algebra and SQL Queries for Functional Requirements

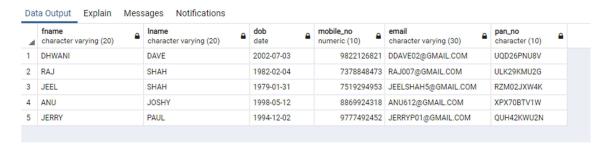
1. Find all the users whose license type is learning license.

Relational Algebra:

 π FNAME,LNAME,DOB,MOBILE_NO,EMAIL,PAN_NO ((σ D_TYPE='LEARNERS' (DRIVING_LICENSE)) \bowtie < DRIVING_LICENSE. USER_ID = USER_DETAILS.USER_ID > USER_DETAILS)

SQL DML Statements:

SELECT FNAME,LNAME,DOB,MOBILE_NO,EMAIL,PAN_NO FROM USER_DETAILS AS U JOIN DRIVING_LICENSE AS D ON (U.USER_ID=D.USER_ID) WHERE D_TYPE='LEARNERS';



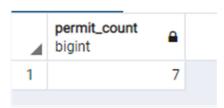
2. Count the number of permits where permit type is "LGV".

Relational Algebra:

r1 <- σ PER_TYPE='LGV'(PERMIT) result <- \mathcal{F} COUNT(*)(r1)

SQL DML Statements:

SELECT COUNT(*) AS PERMIT_COUNT FROM PERMIT WHERE PER_TYPE='LGV';

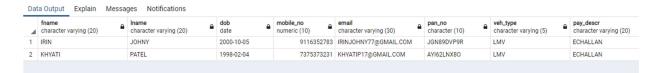


Find all the users whose vehicle type is LMV and who have paid the e-challan.Relational Algebra:

 π FNAME,LNAME,DOB,MOBILE_NO,EMAIL,PAN_NO ((σ VEHICLE.VEH_TYPE='LMV' AND PAYMENT.PAY_DESCR='ECHALLAN'(PAYMENT \bowtie <PAYMENT. USER_ID = USER_DETAILS. USER_ID > \bowtie < VEHICLE_REGISTRATION.USER_ID = USER_DETAILS. USER_ID > \bowtie USER_DETAILS.))

SQL DML Statements:

SELECT FNAME, LNAME, DOB, MOBILE_NO, EMAIL, PAN_NO,VEH_TYPE,PAY_DESCR FROM USER_DETAILS AS U JOIN PAYMENT AS P ON (U.USER_ID=P.USER_ID)
JOIN VEHICLE_REGISTRATION AS V ON (U.USER_ID=V.USER_ID)
WHERE V.VEH_TYPE='LMV' AND P.PAY_DESCR='ECHALLAN';



4. Find all the users having a driving license and whose age is above 18 years. Relational Algebra:

 π FNAME,LNAME,DOB,MOBILE_NO,EMAIL,PAN_NO,AGE(CURRENT_DATE,DOB),D_TYPE ((σ D_TYPE='DRIVING' (DRIVING_LICENSE) \bowtie < DRIVING_LICENSE. USER_ID = USER_DETAILS.USER_ID > USER_DETAILS)

SQL DML Statements:

SELECT FNAME,LNAME,DOB,MOBILE_NO,EMAIL,PAN_NO,AGE(CURRENT_DATE,DOB),D_TYPE FROM USER_DETAILS AS U JOIN DRIVING_LICENSE AS D ON (U.USER ID=D.USER ID) WHERE D TYPE='DRIVING';

4	fname character varying (20) □	Iname character varying (20)	dob date	mobile_no numeric (10)	email character varying (30)	pan_no character (10)	age interval	d_type character varying (50)
1	DHWANI	DAVE	2002-07-03	9822126821	DDAVE02@GMAIL.COM	UQD26PNU8V	18 years 4 mons 22 days	DRIVING
2	RAJ	SHAH	1982-02-04	7378848473	RAJ007@GMAIL.COM	ULK29KMU2G	38 years 9 mons 21 days	DRIVING
3	IRIN	JOHNY	2000-10-05	9116352783	IRINJOHNY77@GMAIL.COM	JGN89DVP9R	20 years 1 mon 20 days	DRIVING
4	ANKIT	DESAI	2001-02-05	8888972477	ANK56@GMAIL.COM	FXW60SZM8R	19 years 9 mons 20 days	DRIVING
5	HARSH	SHAH	1992-05-06	7592459258	HARSH789@GMAIL.COM	PZF57NLW2N	28 years 6 mons 19 days	DRIVING
6	SHUBHI	AGARWAL	1971-09-28	8229757994	SHUBHI275@GMAIL.COM	SEI84PLW4K	49 years 1 mon 27 days	DRIVING
7	SHAN	CHOKSI	1973-10-21	9797988743	SHAN678@GMAIL.COM	ZRH150QT6E	47 years 1 mon 4 days	DRIVING
8	AKANKSHA	SHAH	1972-05-25	9475743182	AKANKSHA22@GMAIL.COM	VBQ24YLK2I	48 years 6 mons	DRIVING
9	MIA	TRIVEDI	1983-03-02	8951481927	MIATRRIVEDI@GMAIL.COM	KZY86NZL5U	37 years 8 mons 23 days	DRIVING
10	NEEL	GUPTA	2001-02-19	8861583648	NEEL008@GMAIL.COM	HQQ02XGA9X	19 years 9 mons 6 days	DRIVING

List the RTO office which has the maximum number of driving license tests scheduled.Relational Algebra:

r1 <- OFFICE_ID FCOUNT(APP_TYPE)->COUNT_DL(APPOINTMENT)
r2 <- (DEPARTMENT ⋈<DEPARTMENT.DEPT_ID=r1.DEPT_ID> r1)
result <- OFFICE_ID,CITY,COUNT(APP_TYPE)->COUNT_DL (OFFICE
⋈<OFFICE_ID=r2.OFFICE_ID> r2)

SQL DML Statements:

SELECT O.OFFICE_ID, CITY, COUNT(APP_TYPE) AS COUNT_DL FROM DEPARTMENT AS D JOIN OFFICE AS O ON (D.OFFICE_ID=O.OFFICE_ID) JOIN APPOINTMENT AS A ON (D.DEPT_ID=A.DEPT_ID) WHERE APP_TYPE='DRIVERS' GROUP BY O.OFFICE_ID;



6. List all the users who have passed the test for a driving license in Surat. Relational Algebra:

 π FNAME,LNAME,DOB,CITY,EMAIL,PAN_NO ((σ LICENSE_RESULT.TEST_RESULT='YES' AND DEPARTMENT.D_TYPE='DRIVING' AND USER_DETAILS.CITY='SURAT'(LICENSE_RESULT)) \bowtie <LICENSE_RESULT.APP_ID = APOINTMENT.APP_ID > APPOINTMENT \bowtie <USER_DETAILS. USER_ID = APPOINTMENT. USER_ID > DEPARTMENT \bowtie < USER_DETAILS.USER_ID= DEPARTMENT.USER_ID>USER_DETAILS)

SQL DML Statements:

SELECT FNAME, LNAME, DOB, CITY, EMAIL, PAN_NO FROM USER_DETAILS AS U JOIN DRIVING_LICENSE AS D ON (U.USER_ID=D.USER_ID) JOIN APPOINTMENT AS A ON (U.USER_ID=A.USER_ID) JOIN LICENSE_RESULT AS T ON (T.APP_ID=A.APP_ID) WHERE T.TEST RESULT='YES' AND D.D TYPE='DRIVING' AND U.CITY='SURAT';

4	fname character varying (20)	Iname character varying (20) □	dob date □	city character varying (50)	email character varying (30)	pan_no character (10)
1	RAJ	SHAH	1982-02-04	SURAT	RAJ007@GMAIL.COM	ULK29KMU2G
2	IRIN	JOHNY	2000-10-05	SURAT	IRINJOHNY77@GMAIL.COM	JGN89DVP9R

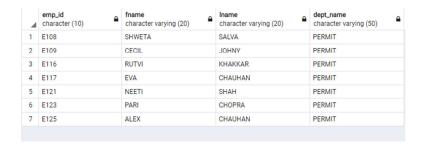
7. Find all the employees working in "PERMIT" department.

Relational Algebra:

 π <EMP_ID,FNAME,LNAME,D.DEPT_NAME>(σ <D.DEPT_NAME='PERMIT'>(ρΕ(EMPLOYEE) \bowtie <E.DEPT_ID=D.DEPT_ID>ρD(DEPARTMENT))

SQL DML Statements:

SELECT EMP_ID, FNAME, LNAME, D.DEPT_NAME FROM EMPLOYEE AS E JOIN DEPARTMENT AS D ON (E.DEPT_ID=D.DEPT_ID) WHERE D.DEPT_NAME='PERMIT';



8. List all the users whose appointments have been cancelled more than two times.
Relational Algebra:

r1 <- σ (APPLICATION.APP_STATUS="CANCELLED")(APPOINTMENT)

r2 <- USER ID \mathcal{F} COUNT(USER DETAILS.USER ID)-> COUNT USER(r1)

 $r3 < -\sigma COUNT_USER > 2 (r2)$

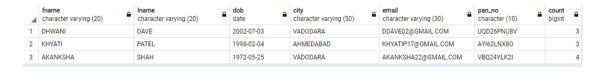
result <- π FNAME,LNAME,DOB,CITY,EMAIL,PAN NO,COUNT USER(\mathcal{F} COUNT (

USER DETAILS.USER ID)->COUNT USER (USER DETAILS) ⋈ (

USER_DETAILS.USER_ID=APPOINTMENT.USER_ID) (r3))

SQL DML Statements:

SELECT FNAME,LNAME,DOB,CITY,EMAIL,PAN_NO,COUNT(U.USER_ID) FROM USER_DETAILS AS U JOIN APPOINTMENT AS A ON (U.USER_ID=A.USER_ID) WHERE A.APP STATUS='CANCELLED' GROUP BY U.USER ID HAVING COUNT(U.USER ID)>2;



Find the users who have not paid e-challan and how many times have they not paid the e-challan.
Relational Algebra:

 σ PAYMENT.USER_ID IS NULL (USER_DETAILS ⋈ < USER_DETAILS.USER_ID=PAYMENT.USER_ID> PAYMENT ⋈ (PAYMENT. USER_ID=ECHALLAN.USER_ID $\mathcal F$ ECHALLAN.USER_ID,COUNT(ECHALLAN.USER_ID) (ECHALLAN))

SQL DML Statements:

SELECT E.USER_ID,COUNT(E.USER_ID) AS E_COUNT FROM USER_DETAILS AS U JOIN PAYMENT AS P ON (U.USER_ID=P.USER_ID) RIGHT JOIN ECHALLAN AS E ON (E.USER_ID=P.USER_ID) WHERE P.USER_ID IS NULL GROUP BY E.USER_ID;

Dat	ta Output	Explain	M	essages	No
4	user_id character ((10)		e_count bigint	<u></u>
1	USID107				2
2	USID109				1
3	USID112				1

10. Find the maximum payment amount received by the Department.

Relational Algebra:

 \mathcal{F} MAX(MAX_COUNT)->MAX_SALARY (DEPT_ID \mathcal{F} DEPT_ID,SUM(AMOUNT)

> MAX_COUNT (PAYMENT)) -> R1

SQL DML Statements:

SELECT MAX(MAX_COUNT) AS MAX_SALARY FROM (SELECT DEPT_ID,SUM(AMOUNT) AS MAX_COUNT FROM PAYMENT GROUP BY DEPT_ID) AS R1;



11. Find the department name where the numbers of employees are more than five.

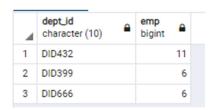
Relational Algebra:

r1 <- EMPLOYEE.DEPT_ID \mathcal{F} COUNT(EMPLOYEE.EMP_ID)-> EMP (EMPLOYEE) r2 <- σ EMP > 5 (r2)

result <- π EMPLOYEE.DEPT_ID,EMP(EMPLOYEE.DEPT_ID \mathcal{F} COUNT(EMPLOYEE.EMP_ID) -> EMP (DEPARTMENT) \bowtie (DEPATMENT.DEPT_ID=EMPLOYEE.DEPT_ID) (r2))

SQL DML Statements:

SELECT E.DEPT_ID,COUNT(E.EMP_ID) AS EMP FROM EMPLOYEE AS E JOIN DEPARTMENT AS D ON (D.DEPT_ID=E.DEPT_ID) GROUP BY E.DEPT_ID HAVING COUNT(E.EMP_ID) > 5;



12. Find the name of the employees whose salary is greater than average salary.

Relational Algebra:

 π EMP_ID, FNAME || '' || LNAME (EMPLOYEE) σ SALARY > ALL (\mathcal{F} AVG(SALARY) (EMPLOYEE)

SQL DML Statements:

SELECT EMP_ID , FNAME | | ' ' | | LNAME AS EMP_NAME FROM EMPLOYEE WHERE SALARY > ALL (SELECT AVG(SALARY) FROM EMPLOYEE);

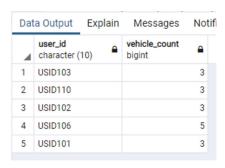
Data	Output Explain Me	essages Notifications
4	emp_id [PK] character (10)	emp_name text
1	E101	MAHIMA DESAI
2	E103	PARTH DAVE
3	E108	SHWETA SALVA
4	E109	CECIL JOHNY
5	E111	ISHIKA SHARMA
6	E112	MEET SHAH
7	E114	PARAM SHAH
8	E116	RUTVI KHAKKAR
9	E119	KHUSHBU KHAKKAR
10	E123	PARI CHOPRA
11	E127	RAHUL SHETTY
12	E131	ANURAG KASHYAP

13. Count the number of users who have registered more than two vehicles. Relational Algebra:

r1 <- VEHICLE_REGISTRATION.USER_ID \mathcal{F} COUNT (VEHICLE_REGISTRATION.USER_ID)-> VH_COUNT (VEHICLE_REGISTRATION) r2 <- σ VH_COUNT > 2 (r1) result <- π VEHICLE_REGISTRATION.USER_ID, COUNT(VEHICLE_REGISTRATION.USER_ID) -> VH_COUNT (USER_DETAILS) \bowtie (USER_DETAILS.USER_ID= VEHICLE_REGISTRATION.USER_ID) (r2))

SQL DML Statements:

SELECT V.USER_ID,COUNT(V.USER_ID) AS VEHICLE_COUNT FROM USER_DETAILS AS U JOIN VEHICLE_REGISTRATION AS V ON (U.USER_ID=V.USER_ID)
GROUP BY V.USER_ID HAVING COUNT(V.USER_ID) > 2;



14. Find the users who have international license which is expiring in the year 2020. Relational Algebra:

 π FNAME,LNAME,DOB,CITY,EMAIL,PAN_NO, DRIVING_LICENSE.D_TYPE, DRIVING_LICENSE.VALIDITY (σ D_TYPE='INTERNATIONAL' AND EXTRACT(YEAR FROM DRIVING_LICENSE.VALIDITY)='2020' (DRIVING_LICENSE) \bowtie (DRIVING_LICENSE.USER_ID=USER_DETAILS.USER_ID) USER_DETAILS)

SQL DML Statements:

SELECT FNAME, LNAME, DOB, CITY, EMAIL, PAN_NO, D.D_TYPE, D. VALIDITY FROM USER_DETAILS AS U JOIN DRIVING_LICENSE AS D ON (U.USER_ID=D.USER_ID) WHERE D_TYPE='INTERNATIONAL' AND EXTRACT(YEAR FROM D.VALIDITY)='2020';



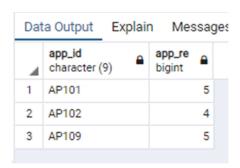
15. Find the users whose appointment has been rescheduled more than 3 times. Relational Algebra:

r1 <- RESECHEDULE.APP_ID $\mathcal F$ COUNT (RESECHEDULE.APP_ID)-> R_COUNT (RESCHEDULE) r2 <- σ R_COUNT > 3 (r1)

result <- π RESECHEDULE.APP_ID,COUNT(RESECHEDULE.APP_ID) -> R_COUNT (USER_DETAILS) \bowtie (USER_DETAILS.USER_ID= APPOINTMENT.USER_ID)(APPOINTMENT) \bowtie (APPOINTMENT.APP_ID=RESECHEDULE.APP_ID) (r2))

SQL DML Statements:

SELECT R.APP_ID,COUNT(R.APP_ID) AS APP_RE FROM USER_DETAILS AS U JOIN APPOINTMENT AS A ON (U.USER_ID=A.USER_ID) JOIN RESECHEDULE AS R ON (A.APP_ID=R.APP_ID) GROUP BY R.APP_ID HAVING COUNT(R.APP_ID) > 3;



16. Find name of user who have paid e-challan at least two times.

Relational Algebra:

r1 <- USER_ID \mathcal{F} COUNT (E_CHALLAN_ID)->E_COUNT (ECHALLAN) r2 <- σ E_COUNT > 1 (r1)

result <- π FNAME,LNAME,DOB,EMAIL,PAN_NO, \mathcal{F} COUNT (ECHALLAN.E_CHALLAN_ID)-> E COUNT ((USER DETAILS) \bowtie (USER DETAILS.USER ID=ECHALLAN.USER ID)r2)

SQL DML Statements:

SELECT FNAME,LNAME,DOB,EMAIL,PAN_NO FROM USER_DETAILS JOIN(SELECT USER_ID,COUNT(E_CHALLAN_ID) FROM ECHALLAN GROUP BY USER_ID HAVING COUNT(E_CHALLAN_ID)>1) AS R1 ON (R1.USER_ID = USER_DETAILS.USER_ID);



17. List all the users who have failed the driving license test three times.

Relational Algebra:

r1 <- σ APP_TYPE = 'DRIVERS' AND APP_STATUS = 'APPROVED' (APPOINTMENT))

 $r2 \leftarrow USER_ID \mathcal{F} COUNT (USER_ID) (r1)$

 $r3 < -\sigma$ COUNT(USER_ID) = 3 (r2)

result $\leftarrow \pi$ USER ID , COUNT(USER ID) (r3)

SQL DML Statements:

SELECT COUNT(USER_ID), USER_ID FROM APPOINTMENT WHERE APP_TYPE = 'DRIVERS' AND APP_STATUS = 'APPROVED' GROUP BY USER_ID HAVING COUNT(USER_ID) = 3:

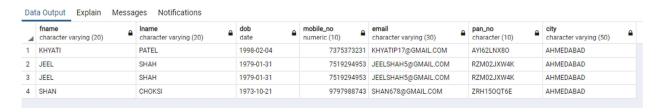


18. List all the users who have applied for vehicle registration for HMV in Ahmedabad. Relational Algebra:

 π FNAME,LNAME,DOB,MOBILE_NO,EMAIL,PAN_NO ((σ VEHICLE.VEH_TYPE='HMV' AND USER_DETAILS.CITY='AHMEDABAD' (VEHICLE_REGISTRATION \bowtie <VEHICLE_REGISTRATION USER_ID = USER_DETAILS.USER_ID > USER_DETAILS.))

SQL DML Statements:

SELECT FNAME,LNAME,DOB,MOBILE_NO,EMAIL,PAN_NO,CITY FROM USER_DETAILS AS U JOIN VEHICLE_REGISTRATION AS V ON (U.USER_ID=V.USER_ID) WHERE VH.VEH_TYPE='HMV' AND U.CITY='AHMEDABAD';

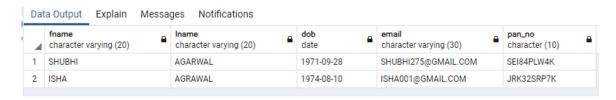


19. Find names of users who have paid E-challan atleast once and belong to the city of Rajkot. Relational Algebra:

r1 <- USER_ID \mathcal{F} COUNT (E_CHALLAN_ID)->E_COUNT (ECHALLAN) r2 <- σ E_COUNT >= 1 (r1) result <- π FNAME,LNAME,DOB,EMAIL,PAN_NO, \mathcal{F} COUNT (ECHALLAN.E_CHALLAN_ID)-> E_COUNT ((USER_DETAILS) \bowtie (USER_DETAILS.USER_ID=ECHALLAN.USER_ID) AND USER_DETAILS.CITY ='RAJKOT') r2)

SQL DML Statements:

SELECT FNAME, LNAME, DOB, EMAIL, PAN_NO FROM USER_DETAILS AS U JOIN (SELECT USER_ID, COUNT (E_CHALLAN_ID) FROM ECHALLAN GROUP BY USER_ID HAVING COUNT (E_CHALLAN_ID) >= 1) AS R1 ON (R1.USER_ID = U.USER_ID) AND U.CITY='RAJKOT';



20. List details of all the user whose name starts with A and has a driving license of type learners.

Relational Algebra:

 π FNAME,LNAME,DOB,CITY, EMAIL,PAN_NO,MOBILE_NO ((σ D_TYPE='LEARNERS' (DRIVING_LICENSE)) \bowtie < DRIVING_LICENSE. USER_ID = USER_DETAILS.USER_ID > USER_DETAILS) WHERE UPPER(FNAME) LIKE 'A%'

SQL DML Statements:

SELECT FNAME, LNAME, DOB, CITY, EMAIL, PAN NO, MOBILE NO from USER DETAILS AS U JOIN

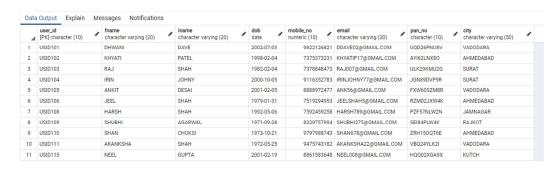


21. List details of all the users who have registered at least one vehicle. Relational Algebra:

r1 <- USER_ID \mathcal{F} COUNT (USER_ID) ->V_COUNT (VEHICLE_REGISTRATION) r2 <- σ V_COUNT >=1 (r1) result <- π *, \mathcal{F} COUNT (VEHICLE_REGISTRATION.LICENSE_PLATE_NO)-> V_COUNT (π USER_ID (USER_DETAILS \bowtie <USER_DETAILS.USER_ID= USER_DETAILS.USER_ID> (USER_DETAILS)) r2)

SQL DML Statements:

SELECT * FROM USER_DETAILS AS U WHERE USER_ID IN (SELECT USER_ID FROM VEHICLE_REGISTRATION WHERE USER_ID=U.USER_ID GROUP BY USER_ID HAVING COUNT(USER_ID)>=1);



22. Find out employee who have salary greater than some employees of department 'ECHALLAN'.

Relational Algebra:

 π * (EMPLOYEE) σ SALARY > SOME(π SALARY (EMPLOYEE) \bowtie <EMPLOYEE.DEPT_ID=DEPARTMENT.DEPT_ID > σ DEPARTMENT.DEPT_NAME='ECHALLAN')

SQL DML Statements:

SELECT * FROM EMPLOYEE WHERE SALARY > SOME (SELECT SALARY FROM EMPLOYEE AS E JOIN DEPARTMENT AS D ON (E.DEPT_ID=D.DEPT_ID) WHERE D.DEPT_NAME='ECHALLAN');

4	emp_id [PK] character (10)	fname character varying (20)	Iname character varying (20)	dob date	mobile_no numeric (10)	email character varying (30)	salary numeric (5)	dept_id character (10)
1	E101	MAHIMA	DESAI	1979-09-15	9746485415	MAHIMA02@GMAIL.COM	50000	DID432
2	E102	VANSHIKA	SHARMA	1983-06-21	9862625142	VSHARMA113@GMAIL.COM	35000	DID137
3	E103	PARTH	DAVE	2002-03-24	9427656614	PARTHDAVE@GMAIL.COM	40000	DID148
4	E104	HARSHIL	PATEL	1998-11-06	9933586131	HARSHIL01@GMAIL.COM	35000	DID432
5	E105	DHYEY	VORA	1977-01-20	7998674728	DHYEY55@GMAIL.COM	30000	DID432
6	E108	SHWETA	SALVA	2068-06-20	8597915469	SHWETA909@GMAIL.COM	50000	DID666
7	E109	CECIL	JOHNY	1987-03-02	7322473113	CECILJ@GMAIL.COM	45000	DID666
8	E110	RENU	ZALA	1995-12-31	8164489359	RENU@GMAIL.COM	30000	DID707
9	E111	ISHIKA	SHARMA	2000-11-12	8768834261	ISHIKA56@GMAIL.COM	50000	DID707
10	E112	MEET	SHAH	1999-07-28	7261542285	MEET90@GMAIL.COM	45000	DID399
11	E113	EVA	SHARMA	1993-03-22	8521869369	EVA11@GMAIL.COM	30000	DID399
12	E114	PARAM	SHAH	1990-01-15	8573641973	PARAMSHAH88@GMAIL.COM	40000	DID101
13	E116	RUTVI	KHAKKAR	1979-09-16	9746485415	RUTVI2@GMAIL.COM	55000	DID578
1/	F117	E//A	CHAIIHAN	1083,06,21	0262625142	EVACOS@GMAIL COM	35000	DID666

4	emp_id [PK] character (10)	fname character varying (20)	Iname character varying (20)	dob date	mobile_no numeric (10)	email character varying (30)	numeric (5)	dept_id character (10)
13	E110	NOTVI	NHANNAN	13/3-03-10	9/40403413	NOT VIZ@GWAIL.COM	33000	סונטוט
14	E117	EVA	CHAUHAN	1983-06-21	9862625142	EVAC05@GMAIL.COM	35000	DID666
15	E119	KHUSHBU	KHAKKAR	1979-09-15	9746485415	KHUSHBU@GMAIL.COM	55000	DID432
16	E120	REETA	CHAUHAN	1983-06-29	9862625142	REETAC@GMAIL.COM	35000	DID432
17	E121	NEETI	SHAH	1983-06-27	9862625142	EVAC05@GMAIL.COM	35000	DID666
18	E123	PARI	CHOPRA	1979-09-16	9746485415	KHUSHBU@GMAIL.COM	55000	DID666
19	E124	SHAKTI	MOHAN	1983-06-26	9862625142	REETAC@GMAIL.COM	35000	DID432
20	E125	ALEX	CHAUHAN	1983-06-22	9862625142	EVAC05@GMAIL.COM	35000	DID666
21	E127	RAHUL	SHETTY	1979-09-15	9746485415	KHUSHBU@GMAIL.COM	55000	DID432
22	E128	VISHNU	CHAUHAN	1983-05-22	9862625142	REETAC@GMAIL.COM	35000	DID399
23	E129	DEV	DESAI	1983-04-21	9862625142	EVAC05@GMAIL.COM	35000	DID399
24	E131	ANURAG	KASHYAP	1979-09-02	9746485415	KHUSHBU@GMAIL.COM	55000	DID432
25	E132	DHRUVIL	SHARMA	1983-02-28	9862625142	REETAC@GMAIL.COM	35000	DID399

23. List details of all the user details who have all the types of driving license Relational Algebra:

 $r1 <-\pi$ USER_DETAILS.USER_ID -> USER_ID, DRIVING_LICENSE. D_TYPE -> D_TYPE ((USER_DETAILS) X DRIVING_LICENSE)

r2 <- r1 - DRIVING LICENSE

 $r3 <- \pi USER_ID(r2)$

Result $\leftarrow \pi$ USER_ID (USER_DETAILS) – r3

SQL DML Statements:

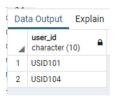
SELECT USER_ID FROM USER_DETAILS

EXCEPT

SELECT USER_ID FROM

(SELECT D.D_TYPE,U.USER_ID FROM USER_DETAILS AS U CROSS JOIN DRIVING_LICENSE AS D EXCEPT

SELECT D_TYPE, USER_ID FROM DRIVING_LICENSE) AS R



24. List all the users whose age is not between 25 to 40 Relational Algebra:

 π FNAME,LNAME,DOB,CITY, EMAIL,PAN_NO,MOBILE_NO,AGE(DOB) (σ AGE(DOB) NOT BETWEEN INTERVAL '25 YEARS' AND INTERVAL '40 YEARS' (USER_DETAILS))

SQL DML Statements:

SELECT FNAME, LNAME, CITY, PAN_NO, MOBILE_NO, AGE(DOB) FROM USER_DETAILS WHERE AGE(DOB) NOT BETWEEN INTERVAL '25 YEARS' AND INTERVAL '40 YEARS';



25. Find the details of users who don't have a permit and are eligible for one.

Relational Algebra:

 $\pi * (\sigma VEHICLE_REGISTRATION.VEH_TYPE='TRANS' AND PER_ID IS NULL (VEHICLE_REGISTRATION))$

SQL DML Statements:

SELECT * FROM VEHICLE_REGISTRATION WHERE VEH_TYPE = 'TRANS' AND PER_ID ISNULL;

