QUERIES

.....

Q1) FIND DEVELOPER WHO HAS NOT PARTICIPATE IN ANY CONTEST OR ASSIGNMENT.

SQL QUERIES

SELECT * FROM DEVELOPER NATURAL JOIN (SELECT DEVELOPER.DEV_ID FROM DEVELOPER LEFT JOIN PARTICIPATE ON(DEVELOPER.DEV_ID=PARTICIPATE.DEV_ID) WHERE PARTICIPATE.DEV_ID IS NULL

UNION

SELECT DEVELOPER.DEV_ID FROM DEVELOPER LEFT JOIN PRACTICE_ASSIGNMENT ON(DEVELOPER.DEV_ID=PRACTICE_ASSIGNMENT.DEV_ID)

WHERE PRACTICE_ASSIGNMENT.DEV_ID IS NULL) AS NONEPARTICIPANT;

RELATIONAL ALGEBRA

$$\begin{split} &\sigma(DEVELOPER)\bowtie P(NONEPARTICIPANT,\Pi_{DEVELOPER.DEV_ID}(\sigma_{PARTICIPATE.DEV_ID} \text{ is null}(\\ &DEVELOPER\bowtie_{DEVELOPER.DEV_ID=PARTICIPATE.DEV_ID} PARTICIPATE)) \ U\\ &\Pi_{DEVELOPER.DEV_ID}(\sigma_{PRACTICE_ASSIGMENT.DEV_ID} \text{ is null}(\ DEVELOPER\\ &\bowtie_{DEVELOPER.DEV_ID=PRACTICE_ASSIGMENT.DEV_ID} \ PRACTICE_ASSIGNMENT)))) \end{split}$$

OUTPUT

4	dev_id [PK] character (7)	d_name character varying (20)	d_email character varying (50)	d_contact_info numeric (10)	d_address character varying (100)	d_country character varying (20)	gender character (1)	dob date
1	DEV0003	KIRAN RUPADJA	kiran@gmail.com	9265999345	8399 Bradford Drive Hartselle,	USA	F	1999-12-28
2	DEV0004	TIRTH PAGEDAR	tirth@gmail.com	9265999456	77 Spring Street Rockaway, N	USA	M	1999-02-07
3	DEV0005	NANDINI MISTRY	nandini@gmail.com	9265999567	9 South Poor House Road Lut	INDIA	F	1999-07-23
4	DEV0006	JAGDISH PATEL	jagdish@gmail.com	9266299123	75 Curran Rd, North Attleboro,	INDIA	M	1996-12-20
5	DEV0008	KHUSHBOO JAIN	khushboo@gmail.com	9265999128	542 Pats Rd, Vass, NC, 28394	SHRILANKA	F	1999-12-13
6	DEV0009	ANIKET MISTRY	aniketM@gmail.com	9898999123	4580 Vintage Rd, Lilesville, NC	HONGKONG	M	1997-03-20
7	DEV0010	ROHIT TANTI	rohittanti@gmail.com	9292324561	7888 Genery Trl, Mc Calla, AL,	NEPAL	M	1999-06-20
8	DEV0011	POOJA SHAH	poojashah@gmail.com	9265999222	7103 Trimstone Dr, Pasadena,	INDIA	F	1999-12-08
9	DEV0012	AESHA KAYESTHA	aeshak@gmail.com	9253999123	2346 Little River Rd, Henderso	US	F	1999-09-09
10	DEV0013	BANDISH SHETH	bandish@gmail.com	9213999126	1162 Lower River Rd, Covingt	INDIA	М	1992-12-20

Q2) FIND DEVELOPER WHO HAS NOT PARTICIPATE IN CONTEST BUT HAS PARTICIPATE IN ASSIGNMENT.

SQL QUERIES

SELECT * FROM DEVELOPER NATURAL JOIN (SELECT DEVELOPER.DEV_ID FROM DEVELOPER LEFT JOIN PARTICIPATE ON(DEVELOPER.DEV_ID=PARTICIPATE.DEV_ID)

WHERE PARTICIPATE.DEV_ID IS NULL

EXCEPT

SELECT DEVELOPER.DEV_ID FROM DEVELOPER LEFT JOIN PRACTICE_ASSIGNMENT ON(DEVELOPER.DEV_ID=PRACTICE_ASSIGNMENT.DEV_ID)

WHERE PRACTICE_ASSIGNMENT.DEV_ID IS NULL) AS ONLYCONTEST;

RELATIONAL ALGEBRA

Data Output Explain Messages Notifications

$$\begin{split} &\sigma(DEVELOPER)\bowtie P(ONLYCONTEST,\Pi_{DEVELOPER.DEV_ID}(\sigma_{PARTICIPATE.DEV_ID}\ is\ \text{null}(\\ &DEVELOPER\bowtie_{DEVELOPER.DEV_ID=PARTICIPATE.DEV_ID}\ PARTICIPATE\))\ -\\ &\Pi_{DEVELOPER.DEV_ID}(\sigma_{PRACTICE_ASSIGMENT.DEV_ID}\ is\ \text{null}(\ DEVELOPER\\ &\bowtie_{DEVELOPER.DEV_ID=PRACTICE_ASSIGMENT.DEV_ID}\ PRACTICE_ASSIGNMENT))\) \end{split}$$

OUTPUT

1 DEV0004	TIRTH PAGEDAR	tirth Oamail aam	0045000454				
	11111111110257111	tirth@gmail.com	9265999456	77 Spring Street Rockaway, NJ 0	USA	M	1999-02-07
2 DEV0005	NANDINI MISTRY	nandini@gmail.com	9265999567	9 South Poor House Road Luthe	INDIA	F	1999-07-23

Q3) FIND THE QUESTION WHO HAS MAXIMUM NUMBER OF TEST CASE.

SQL QUERIES

SELECT * FROM QUESTION NATURAL JOIN (SELECT QUE_ID,COUNT(QUE_ID) FROM TESTCASE GROUP BY QUE_ID ORDER BY QUE_ID DESC)AS TCASE LIMIT 1;

RELATIONAL ALGEBRA

 $\sigma_{\text{LIMIT 1}}(\text{QUESTION} \bowtie P(\text{TCASE}, \text{QUE_ID}\mathfrak{I}_{\text{QUE_ID},\text{COUNT_ID}}(\text{TESTCASE}) \text{ ORDER BY QUE_ID DESC }))$

NOTE: RELATIONAL ALGEBRA NOT PROVIDE ANY WAY TO EXPRESS ORDER BY AND LIMIT.

Dat	t a Output Explain I	Messages Notifications				
4	que_id [PK] character (7)	title character varying (40)	path character varying (40)	difficulty_level character varying (10)	total_score integer	skill_id character (7)
1	QUE0005	Number Line Jumps	Questions/JS/QUE0005.html	BEGINNER	10	SKI0005

Q4) FIND JAVA PROGRAMMER WHO HAVE AGE GREATER THAN 20 AND TAKE PART IN CONTEST ORGANIZED BY BIRLA.

SQL QUERIES

SELECT * FROM DEVELOPER NATURAL JOIN CONTAINS NATURAL JOIN SKILL

NATURAL JOIN

(SELECT DEV_ID FROM COMPANY JOIN CONTEST ON (COMPANY.COM_ID=CONTEST.COM_ID) JOIN PARTICIPATE ON(PARTICIPATE.CON_ID=CONTEST.CON_ID) WHERE COMPANY.C_NAME='BIRLA INDUSTRIES')AS CONTESTBYBRILA WHERE S_NAME='JAVA' AND AGE(DOB) >'20 YEAR'::INTERVAL

RELATIONAL ALGEBRA

σ (DEVELOPER ⋈ CONTAINS ⋈ SKILL ⋈ PARTICIPATE P (CONTESTBY BIRLA, Π_{DEV_ID} (σ NAME='JAVA' AND AGE(DOB)>'20 YEAR':

INTREVAL(COMPANY ⋈ COMPANY.COM_ID=CONTEST.COM_IDCONTEST.COM_ID=PARTICIPATE.CO NIDPARTICIPATE))))

OUTPUT

dev_id character (7) d_name character varying (20) d_email character varying (50)	ш.	contact_info meric (10)	Δ –	_address	d_country	gender
	IIII	illelic (10)	cna	haracter varying (100)	character varying (20)	character
1 DEV0002 KARAN PATEL karan@gmail.com		92659992	234 356	56 S. Lookout St. Kissimmee, FL	INDIA	M

Q5) LIST OF ASSIGNMENT DETAIS WHO ARE CREATED BY PARTICULAR ADMIN WHO'S NAME START WITH EITHER H OR P.

SQL QUERIES

SELECT * FROM ASSIGNMENT JOIN ADMIN ON(ASSIGNMENT.ADM_ID=ADMIN.ADM_ID) WHERE UPPER(ADMIN.A_NAME) SIMILAR TO '(P|H)%'

RELATIONAL ALGEBRA

σ upper(admin.a_name) similar to

'(P|H)%'(ASSIGNMENT ⋈ ASSIGNMENT.ADM ID=ADMIN.ADM IDADMIN)

Dat	t a Output Explair	n Messages Notification	ons						
4	assi_id character (7)	a_name character varying (50)	a_description character varying (100)	a_status character varying (10)	skill_id character (7)	adm_id character (7)	creat times	adm_id character (7)	a_name character varying (20)
1	AS00001	ARRAY IN C++	Initializing dynamically allocat	PUBLISH	SKI0001	ADM0001	20	ADM0001	PRUTHVIN JHAVERI
2	AS00002	ARRAY IN C	Initializing dynamically allocat	UN-PUBLISH	SKI0002	ADM0001	20	ADM0001	PRUTHVIN JHAVERI
3	AS00003	INHERITANCE IN JAVA	Types of inheritance in JAVA	PUBLISH	SKI0003	ADM0002	20	ADM0002	HIMANSHU JOSHI
4	AS00005	SETS IN PYTHON	Adding Elements to a Set,Acc	UN-PUBLISH	SKI0004	ADM0005	20	ADM0005	PARTH LAKKAD

Q6) FIND DEVLOPER WHO GOT RECUIRTED IN CONTEST NAME 'COMPETETIVE TEST BY AMAZON'.

SQL QUERIES

SELECT * FROM DEVELOPER NATURAL JOIN (SELECT DEV_ID FROM CONTEST JOIN PARTICIPATE ON(CONTEST.CON_ID=PARTICIPATE.CON_ID) WHERE C_NAME='COMPETETIVE TEST BY AMAZON' AND IS_RECRUITED='T') AS WINERLIST

RELATIONAL ALGEBRA

σ (DEVELOPER \Join P(WINERLIST, Π_{DEV_ID} ($\sigma_{C_NAME="COMPETETIVE TEST BY AMAZON" AND IS RECRUIRED="T")(CONTEST <math>\Join_{CONTEST.CON\ ID=PARTICIPATE.CON\ ID}PARTICIPATE))))$

OUTPUT

Jai	a Output Explain	Messages Notifications						
4	dev_id [PK] character (7)	d_name character varying (20)	d_email character varying (50)	d_contact_info numeric (10)	d_address character varying (100)	d_country character varying (20)	gender character (1)	dob date
1	DEV0003	KIRAN RUPADJA	kiran@gmail.com	9265999345	8399 Bradford Drive Hartselle,	USA	F	1999-12-2

Q7) FIND DEVEVLOPER AND ITS JOB TITLE WHO WAS EVER RECRUITED

SQL QUERIES

SELECT JOB_TITLE, DEVELOPER.* FROM DEVELOPER NAUTRAL JOIN PARTICIPATE NATURAL JOIN JOB_INFORMATION WHERE IS_RECRUITED='T'

RELATIONAL ALGEBRA

 $\Pi_{JOB_TITLE,DEVELOPER.*}(\sigma_{IS_RECRUITED='T'}(DEVELOPER))$ $\bowtie PARTICIPATE \bowtie JOB_INFORMATION))$

OUTPUT

Data Output Explain Messages Notifications

4	job_itle character varying (20)	dev_id character (7)	d_name character varying (20)	d_email character varying (50) △	d_contact_info numeric (10)	d_address character varying (100)	d_country character varying (20)	gender characte
1	Database Manager	DEV0002	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee, F	INDIA	M
2	Software Developer	DEV0003	KIRAN RUPADJA	kiran@gmail.com	9265999345	8399 Bradford Drive Hartselle, A	USA	F

Q8) FIND DEVELOPER WHO WAS RECURETED AT LEAST 2 TIME.

SQL QUERIES

SELECT * FROM DEVELOPER NATURAL JOIN (SELECT DEV_ID FROM PARTICIPATE WHERE IS_RECRUITED='T' GROUP BY DEV_ID HAVING COUNT(*)>1)AS ATLEST2

RELATIONAL ALGEBRA

 $\begin{array}{l} \sigma \; (DEVELOPER \bowtie P(ATLEAST2, \sigma \;_{IS_RECRUITED='T'\;AND} \\ count(*)>1(Dev_iD \mathfrak{I}_{DEV_ID}(PARTICIPATE)))) \end{array}$

OUTPUT

Dat	ta Output Explain	Messages Notifications						
4	dev_id [PK] character (7)	d_name character varying (20)	d_email character varying (50)	d_contact_info numeric (10)	d_address character varying (100)	d_country character varying (20)	gender character (1)	dob date
1	DEV0002	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee	INDIA	М	1999-12-05
2	DEV0003	KIRAN RUPADJA	kiran@gmail.com	9265999345	8399 Bradford Drive Hartselle,	USA	F	1999-12-28

Q9) FIND DEVELOPER WHO GOT HIGHEST OFFER(SALARY).

SQL QUERIES

SELECT * FROM DEVELOPER NATURAL JOIN PARTICIPATE NATURAL JOIN JOB_INFORMATION WHERE IS_RECRUITED='T' AND SALARY=(SELECT MAX(SALARY) AS MAX FROM JOB_INFORMATION)

RELATIONAL ALGEBRA

 $R1:= \mathfrak{I}_{MAX(SALARY)\rightarrow MAX}(JOB_INFORMATION)$

 σ is_recuirted='t' and salary=R1.MAX

(DEVELOPER ⋈ PARTICIPATE ⋈ JOB_INFORMATION)

Da	ta Ou	tput Explain	M	essages Notific	eations					
4	crea time	con_id character (10)	-	dev_id character (7)	d_name character varying (20)	d_email character varying (50) □	d_contact_info numeric (10)	d_address character varying (100)	d_country character varying (20)	gender character (1)
1	2	CON0002		DEV0002	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee	INDIA	М

Q10) FIND THE SKILL WHICH IS MOST ACQUIRED BY DEVELOPER (C,C++,JAVA,ETC);

SQL QUERIES

SELECT * FROM SKILL NATURAL JOIN (SELECT SKILL_ID FROM CONTAINS GROUP BY SKILL_ID ORDER BY COUNT(*) DESC LIMIT 1)AS ASKILL

RELATIONAL ALGEBRA

SKILL™P(ASKILL, SKILL_ID SKILL_ID ORDER BY SKILLID DESC LIMIT 1)

NOTE: ORDER BY AND LIMIT ARE NOT ABLE TO EXPRESS IN REALTIONAL ALGEBRA

OUTPUT

Data	a Output Explain	Messages Notifications		
4	skill_id [PK] character (7)	s_name character varying (20)	s_description character varying (500)	improvement character varying (500)
1	SKI0001	C++	C++ is a middle-level program	WORK ON Switch statement,F

Q11) FIND DEVELOPER ID WHO HAS PARTICIPATED IN ALL THE CONTEST ORGANIZED BY AMAZON

SQL QUERIES

SELECT DEV_ID FROM PARTICIPATE

EXCEPT

SELECT DEV_ID FROM

(SELECT CONTEST.CON_ID,PARTICIPATE.DEV_ID FROM PARTICIPATE CROSS JOIN CONTEST JOIN COMPANY ON (CONTEST.COM_ID=COMPANY.COM_ID) WHERE COMPANY.C_NAME='AMAZON'

EXCEPT

SELECT CON_ID, DEV_ID FROM PARTICIPATE) AS R1

RELATIONAL ALGEBRA

 $\Pi_{DEV_ID}(PARITICIPATE) - \Pi_{DEV_ID}(P(R1,\Pi_{CONTEST.CON_ID,PARTICIPATE.DEV_ID}(\sigma_{COMPANY.C_NAME='AMAZON'}(PARTICIPATEXCONTEST.COM_ID=COMPANY.COM_ID=COMPANY) - \Pi_{CON_ID,DEV_ID}(PARTICIPATEX))))$

OUTPUT



Q12) FIND DEVELOPER WHO ACQUIRE ALL THE SKILL

SQL QUERIES

SELECT DEV_ID FROM CONTAINS

EXCEPT

SELECT DEV_ID FROM

(SELECT DEV_ID,SKILL_ID FROM DEVELOPER CROSS JOIN SKILL

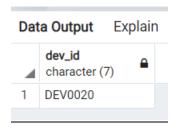
EXCEPT

SELECT DEV_ID,SKILL_ID FROM CONTAINS)AS R1;

RELATIONAL ALGEBRA

 $\Pi_{DEV_ID}(CONTAINS) - \Pi_{DEV_ID}(P(R1,\Pi_{DEV_ID,SKILL_ID}(DEVELOPER~X~SKILL) - \Pi_{DEV_ID,SKILL_ID}(CONTAINS)))$

OUTPUT



Q13) FIND DEVELOPER WHO HAVE PARTICIPATED IN ALL THE ASSIGNMENT

SQL QUERIES

SELECT DEV_ID FROM PRACTICE_ASSIGNMENT

EXCEPT

SELECT DEV_ID FROM

(SELECT DEV_ID,ASSI_ID FROM ASSIGNMENT,DEVELOPER

EXCEPT

SELECT DEV_ID,ASSI_ID FROM PRACTICE_ASSIGNMENT)AS R1

RELATIONAL ALGEBRA

 $\Pi_{DEV_ID}(PRACTICE_ASSIGNMENT) - \Pi_{DEV_ID}(P(R1,\!\Pi_{DEV_ID,ASSI_ID}(ASSIGNMENT~X~DEVELOPER) - \Pi_{DEV_ID,ASSI_ID}(PRACTICE_ASSIGNMENT)~))$

OUTPUT



.....

O14) FIND THE TOP 10 DEVELOPERS WHO HAS ATTENDED MAXIMUM NUMBER OF CONTEST

SQL QUERIES

SELECT DEV_ID,COUNT(CON_ID) AS NUM FROM PARTICIPATE GROUP BY DEV_ID ORDER BY DEV_ID

DESC LIMIT 10

RELATIONAL ALGEBRA

σ(ORDER BY DEV_ID DESC AND LIMIT 10 (DEV_IDℑDEV_ID,COUNT(CON_ID)->NUM(PARTICIPATE)

OUTPUT

4	dev_id character varying □	num integer	a
1	DEV0001	10	0
2	DEV0002	9	9
3	DEV0003	8	8
4	DEV0004	6	6
5	DEV0005	25	5
6	DEV0025	- 2	2
7	DEV0006	2	2
8	DEV0016	-	1
9	DEV0024		1
10	DEV0014		1

Q15) SELECT DEVELOPER WHO WAS OFFERED SALARY MORE THAN AVG

SQL QUERIES

SELECT DEVELOPER.* FROM DEVELOPER NATURAL JOIN PARTICIPATE NATURAL JOIN JOB_INFORMATION WHERE SALARY >=

(SELECT AVG(SALARY) AS AVG FROM JOB_INFORMATION NATURAL JOIN PARTICIPATE WHERE IS_RECRUITED='T')

RELATIONAL ALGEBRA

 $R1 = \sigma_{\text{IS RECRUITED='T'}} (\mathfrak{I}_{\text{AVG(SALARY)-> AVG}} (\text{JOB_INFORMATION} \bowtie \text{PAERTICIPATE}))$

 $\sigma_{SALARY>=R1.AVG}(DEVELOPER \bowtie PARTICIPATE \bowtie JOB_INFORMATION)$

	ta Output Explain	Messages Notifications	1					
4	dev_id [PK] character (7)	d_name character varying (20)	d_email character varying (50)	d_contact_info numeric (10)	d_address character varying (100)	d_country character varying (20)	gender character (1)	dob date
1	DEV0002	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee, F	INDIA	М	1999-12-05

Q16) SELECT TOP 5 DEVELOPER BASE ON SCORE

SQL QUERIES

SELECT * FROM DEVELOPER NATURAL JOIN PROFILE ORDER BY TOATAL_SCORE DESC LIMIT 5;

RELATIONAL ALGEBRA

O ORDER BY TOATAL_SCORE AND LIMIT 50 (DEVELOER ⋈ PROFILE)

OUTPUT

Data Output Explain Messages Notifications

4	dev_id character (7)	create_at timestamp with time zone	d_name character varying (20)	d_email character varying (50)	d_contact_info numeric (10)	d_address character varying (100)	d_country character varying (20)
1	DEV0004	2020-10-09 14:31:12.830583+05:30	TIRTH PAGEDAR	tirth@gmail.com	9265999456	77 Spring Street Rockaway, NJ 0	USA
2	DEV0001	2020-10-09 14:31:12.830583+05:30	RAJ PATEL	rajpatel@gmail.com	9265999123	7741 Bellevue Drive Auburndale,	INDIA
3	DEV0002	2020-10-09 14:31:12.830583+05:30	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee, F	INDIA
4	DEV0005	2020-10-09 14:31:12.830583+05:30	NANDINI MISTRY	nandini@gmail.com	9265999567	9 South Poor House Road Luthe	INDIA
5	DEV0003	2020-10-09 14:31:12.830583+05:30	KIRAN RUPADJA	kiran@gmail.com	9265999345	8399 Bradford Drive Hartselle, A	USA

Q17) FIND THE NAME OF DEVELOPER WHO HAS ATTENDED HIGHEST NUMBER OF ASSIGNMENT

SQL QUERIES

SELECT D_NAME FROM DEVELOPER AS D JOIN

(SELECT DEV_ID,COUNT(ASSI_ID) AS NUM FROM PRACTICE_ASSIGNMENT GROUP BY DEV_ID ORDER BY NUM DESC LIMIT 1)

AS C1 ON D.DEV ID=C1.DEV ID

RELATIONAL ALGEBRA

$$\label{eq:condition} \begin{split} \Pi D_NAME(\rho(D,DEVELOPER) \bowtie D.DEV_ID=C1.DEV_ID \; \rho(C1,\;\sigma\;ORDER\;BY\;NUM\;DESC\;AND\;LIMIT\;1\\ (DEV_ID\Im DEV_ID,COUNT(ASSI_ID)(PRACTICE_ASSIGNMENT)))' \end{split}$$

OUTPUT



Q18) SELECT MOST DIFFICULT QUESTION BASED ON PEOPLE ATTEMP AND UNABLE TO SOLVE IT

SQL QUERIES

SELECT QUE_ID,COUNT(QUE_ID) FROM

(SELECT QUE_ID FROM ATTEND_QUE_ASSIGNMENT WHERE STATUS='UNSOLVED'

UNION

SELECT QUE_ID FROM ATTEND_QUE_CONTEST WHERE STATUS='UNCOMPLETE') AS R1 GROUP BY QUE_ID ORDER BY COUNT(QUE_ID) LIMIT 1;

RELATIONAL ALGEBRA

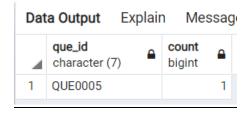
 σ limit 1 and order by Que_id(Que_id \mathfrak{I}_{QUE} Que_id,Count(Que_id)(P(R1,

 $\Pi_{QUE_ID}(\Sigma_{STATUS='UNSOLVED'}(ATTEND_QUE_ASSIGNMENT))$

U

 $\Pi_{QUE_ID}(\sigma_{STATUS='UNCOMPLETE'}(ATTEND_QUE_CONTEST))\)))$

OUTPUT



Q19) FIND DEVELOPER WHO GOT ALTEAST ONE CIRTIFICATE

SQL QUERIES

SELECT DEV_ID,COUNT(DEV_ID) FROM CERTIFICATE GROUP BY DEV_ID HAVING COUNT(DEV_ID) >= 1;

RELATIONAL ALGEBRA

 $\sigma_{COUNT(DEV_ID)>=1} \text{ and order by dev_id} \\ (\text{Dev_id} \\ \mathcal{I}_{DEV_ID,COUNT(DEV_ID)} \\ (\text{CERTIFICATE}))$

OUTPUT

Data Output dev id count character varying integer DEV0001 10 DEV0016 8 7 3 DEV0003 4 DEV0004 6 5 5 DEV0002 DEV0005 4 DEV0006 3 DEV0025 2 DEV0024 2 DEV0014

Q20) FIND COMPANY WHO RECUIRETED MAX NUMBER OF DEVELOPER

SQL QUERIES

SELECT COM_ID FROM PARTICIPATE JOIN CONTEST ON (PARTICIPATE.CON_ID=CONTEST.CON_ID) WHERE IS_RECRUITED='T'

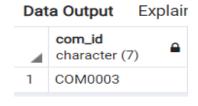
GROUP BY COM_ID ORDER BY COUNT(COM_ID) DESC LIMIT 1;

RELATIONAL ALGEBRA

σ is_recruited='t' and order by com_id and limit 1(com_id³com_id-(PARTICIPATE⋈PARTICIPATE.CON_iD=CONTEST.CON_idCONTEST))

NOTE: RELATIONAL ALGEBRA NOT PROVIDE ANY WAY TO EXPRESS ORDER BY AND LIMIT.

OUTPUT



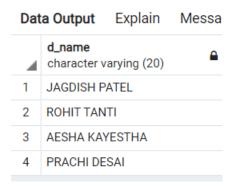
Q21) LIST OF THE DEVELOPER WHO HAVE CURRENT_EDUCATION IS 'MSC IT';

SQL QUERIES

SELECT D_NAME FROM DEVELOPER NATURAL JOIN PROFILE WHERE CURRENT_EDUCATION='MSCIT';

RELATIONAL ALGEBRA

 $\pi_{d_name}(\sigma_{CURRENT_EDUCATION='MSCIT'}(DEVELOPER \bowtie PROFILE))$



Q22) LIST NAME OF DEVELOPER WHO HAS ATTEND TEST CASE WHO'S OUTPUT LIKE '206';

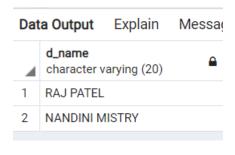
SQL QUERIES

SELECT D_NAME FROM DEVELOPER NATURAL JOIN ATTEND_TESTCASE_ASSIGNMENT NATURAL JOIN TESTCASE WHERE OUTPUT='206';

RELATIONAL ALGEBRA

 $\Pi_{D \text{ NAME}}(\sigma_{\text{OUTPUT}='206'}(\text{ATTEND_TESTCASE_ASSIGNMENT} \bowtie \text{TESTCASE}))$

OUTPUT



Q23) COUNT THE NUMBER OF DEVELOPER WHICH ATTEMPT MAXIMUM NUMBER OF QUESTION;

SQL QUERIES

SELECT DEV_ID, COUNT(DEV_ID) FROM DEVELOPER NATURAL JOIN ATTEND_QUE_ASSIGNMENT GROUP BY DEV_ID ORDER BY COUNT(DEV_ID) DESC LIMIT 1;

RELATIONAL ALGEBRA

 $\sigma_{LIMIT\ 1} \ and \ order\ by \\ Dev_ID(Dev_ID \ Dev_ID,COUNT(Dev_ID) (DEVELOPER \bowtie ATTEND_QUE_ASSIGNMENT))$



Q24) NAME OF DEVELOPER WHO GOT THE MAXIMIM NUMBER OF SCORE IN QUESTION;

SQL QUERIES

SELECT D_NAME, MAX(TOTAL_SCORE) FROM DEVELOPER NATURAL JOIN CONTAINS NATURAL JOIN QUESTION GROUP BY TOTAL_SCORE ,D_NAME ORDER BY MAX(TOTAL_SCORE) DESC LIMIT 1;

RELATIONAL ALGEBRA

 $\sigma_{\text{ LIMIT 1 AND ORDER BY MAX(TOTAL_SCORE)}} \text{ } \text{DEVELOPER} \bowtie \text{CONTONS} \bowtie \text{QUESTION}))$

OUTPUT

ı	Dat	Data Output Expla		Messa	ges 1	Votific
	4	d_name character v	rarying (20)	•	max integer	<u> </u>
	1	TIRTH PAG	EDAR			50

Q25) NAME OF THE DEVELOPER WHO ATTEMPTED THE QUESTION WHICH HAD DIFICULTY LEVEL AS MODERATE AND TITLE AS INHERITED CODE.

SQL QUERIES

SELECT D_NAME, Q.DIFFICULTY_LEVEL FROM DEVELOPER AS D JOIN CONTAINS AS C ON(D.DEV_ID = C.DEV_ID) JOIN QUESTION AS Q ON(Q.SKILL_ID = C.DEV_ID) WHERE DIFFICULTY_LEVEL='MODERATE' AND TITLE='INHERITED CODE';

RELATIONAL ALGEBRA

πd_NAME, DIFFICULTY_LEVEL (ODIFFICULTY_LEVEL='MODERATE' AND TITLE='INHERITED CODE' (DEVELOPER CONTAINS QUESTION))

OUTPUT

Data Output Explain Messages Notifications

4	dev_id character (7)	d_name character varying (20)	d_email character varying (50) □	d_contact_info numeric (10)	d_address character varying (100) △	d_country character varying (20)	gender character (1)
1	DEV0002	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee, FL	INDIA	M

Q26) TOP 5 DEVELOPER WHO GOT MAXIMUM SCORE AND BELONG FROM INDIA

SQL QUERIES

SELECT D_NAME, MAX(P.TOATAL_SCORE) FROM DEVELOPER AS D JOIN PROFILE AS P ON(P.DEV_ID = D.DEV_ID) WHERE D.D_COUNTRY='INDIA' GROUP BY D_NAME, D_COUNTRY, P.TOATAL_SCORE ORDER BY P.TOATAL_SCORE DESC LIMIT 5;

RELATIONAL ALGEBRA

σlimit 5 and d_country='india' and order by desc
(d_name,d_country,toatal_score

J_name,max(total_score))

OUTPUT

Dat	a Output	Explain	Messa	ges	Notifica
4	d_name character v	arying (20)	•	max intege	er 🖴
1	JITARTH sh	nah			2678
2	PRACHI DE	SAI			2624
3	HITDHARM	DOSHI			2623
4	RAJ PATEL				2565
5	JAGDISH P	ATEL			2523

.....

Q27) LIST DEVELOPER DETAILS WHO HAVE TRAVELLING AS A HOBBY AND HAVE AT LEAST 1000 POINTS IN JAVA.

SQL Queries

SELECT * from DEVELOPER NATURAL JOIN HOBBY NATURAL JOIN PROFILE WHERE LOWER(H_NAME) = 'traveling' AND TOATAL_SCORE > 1000;

• Relational Algebra

σ <LOWER(H_NAME)="TRAVELING" AND TOATAL_SCORE >1000 >(DEVLOPER ⋈ HOBBY ⋈ PROFILE)



Q28) LIST DEVELOPERS WHO WERE RECRUITED BY "BIRLA INDUSTRIES" AFTER 2018

• SQL Queries

SELECT * from DEVELOPER NATURAL JOIN PARTICIPATE NATURAL JOIN CONTEST JOIN COMPANY ON CONTEST.COM_ID=COMPANY.COM_ID WHERE COMPANY.C_NAME ='BIRLA INDUSTRIES' AND PARTICIPATE.CREATE_AT >= '2018-01-01 14:31:12.830583+05:30' AND PARTICIPATE.IS_RECRUITED='T'

• Relational Algebra

 σ < COMPANY.C_NAME ='BIRLA INDUSTRIES' AND PARTICIPATE.CREATE_AT >= '2018-01-01 00:00:00+05:30' AND PARTICIPATE.IS_RECRUITED='T'' > (DEVELOPER \bowtie PARTICIPATE \bowtie COMPANY)

• Output



Q29) LIST THE BEST PERFORMERS IN A CONTEST WHOSE ID IS CON003.

SQL Queries

SELECT * FROM PARTICIPATE WHERE CON_ID='CON0003' ORDER BY SCORE DESC LIMIT 1;

• Relational Algebra

 $\sigma_{\text{con_id='con0003'}}$ and order by score desc and limit 1 (participate)

• Output

Dat	ta Output Explain		Лe	essages Notifications				
4	dev_id [PK] characte	er (7)	•	con_id [PK] character (10)	ore teger	(*)	is_recruited boolean	(4)
1	DEV0003			CON0003		450	true	

Q30) LIST OUT THE CONTEST, WHICH HAD ALLOTTED MAXIMUM TIME DURATION.

SQL Queries

SELECT * FROM CONTEST WHERE (END_TIME_DATE-START_TIME_DATE) = (SELECT MAX(END_TIME_DATE-START_TIME_DATE) = (FROM CONTEST);

• Relational Algebra

R1=\(\mathfrak{I}\) MAX(END_TIME_DATE - START_TIME_DATE)->MAX

 $\sigma_{END_TIME-START_TIME_DATE=R1.MAX}(CONTEST)$

• Output

Dat	a Output Explain M	essages Notifications					
1	con_id [PK] character (10)	c_name character varying (50)	start_time_date timestamp without time zone	end_time_date timestamp without time zone	c_status character varying (10)	adm_id character (7)	com_id character (7)
1	CON0001	COMPETETIVE TEST BY TATA	2020-09-15 15:00:00	2020-09-15 18:00:00	PUBLISH	[null]	COM0001
2	CON0002	COMPETETIVE TEST BY BIR	2020-08-19 15:00:00	2020-08-19 18:00:00	PUBLISH	[null]	COM0002
3	CON0003	COMPETETIVE TEST BY AM	2020-12-28 15:00:00	2020-12-28 18:00:00	UN-PUBLISH	[null]	COM0003

Q31) LIST THE DEVELOPERS WHO ARE FROM BCA AND GET RECRUITED BY ANY COMPANY.

• SQL Queries

SELECT D_NAME FROM DEVELOPER D1 JOIN

(SELECT DEV_ID,QUALIFICATION FROM PROFILE)AS P1 ON D1.DEV_ID=P1.DEV_ID JOIN

(SELECT DEV_ID,CON_ID,IS_RECRUITED FROM PARTICIPATE)AS P2 ON D1.DEV_ID=P2.DEV_ID

AND P2.IS_RECRUITED='T' AND P1.QUALIFICATION='BCA'

• Relational Algebra

 Π P(D1, σ D_name(Developer)) \bowtie <P1.Dev_ID=D1.Dev_ID>
P(P1, σ Dev_ID, Qualification='BCA'(profile) \bowtie <D1.Dev_ID=P2.Dev_ID>
P(P2, σ Dev_ID, is_recruited='T'(participate)) σ

• Output

4	d_name character varying (20)
1	RAJ PATEL
2	KIRAN RUPADJA

.....

Q32) LIST THE NON-IT COMPANIES WHO PROVIDE PACKAGE ABOVE 9 LAKH.

SQL Queries

SELECT C_NAME,COMPANY_TYPE FROM COMPANY AS C JOIN

(SELECT CON_ID,COM_ID FROM CONTEST)AS CON1 ON C.COM_ID=CON1.COM_ID JOIN

(SELECT SALARY,CON_ID FROM JOB_INFORMATION) AS J ON J.CON_ID=CON1.CON_ID

WHERE COMPANY TYPE='NON-IT'

AND CON1.COM_ID IS NOT NULL AND J.SALARY>900000

• Relational Algebra

 $\prod P(C, \sigma C_name, Company_type='non-IT'(Company))$

 $\bowtie < C.COM_ID = CON1.COM_ID >$

 $P(CON1,\sigma CON_ID,COM_ID (Contest) \bowtie < J.CON_ID = CON1.CON_ID >$

P(J,σ SALARY>900000,CON_ID, COMPANY_TYPE='NON_IT' (Job_Information))

Output



Q33) LIST THE DEVELOPERS WHO IS RECRUITED AS A 'DATABASE MANAGER' AND ALSO HAVE A SKILL OF C++ OR JAVA.

• SQL Queries

SELECT DISTINCT D_NAME FROM DEVELOPER AS D JOIN (SELECT DEV_ID,SKILL_ID FROM CONTAINS WHERE SKILL_ID='SKI0001' or SKILL_ID='SKI0002') AS C1 ON D.DEV_ID=C1.DEV_ID JOIN (SELECT DEV_ID,CON_ID,IS_RECRUITED FROM PARTICIPATE WHERE IS_RECRUITED='T') AS P ON P.DEV_ID=C1.DEV_ID JOIN (SELECT JOB_ITLE,CON_ID FROM JOB_INFORMATION WHERE JOB_ITLE='Database Manager') AS J ON P.CON_ID=J.CON_ID

• Relational Algebra

 $\Pi P(D, \sigma D_name(DEVELOPER))$

 $\bowtie < D.DEV_ID = C1.DEV_ID >$

P(C1, \sigma SKILL ID='SKIOO1' OR SKILL- ID='SKIO02' (Contest) \in < P.DEV ID=C1.DEV ID>

 $P(P,\sigma DEV_ID,CON_ID,IS_RECRUITED='T'(PARTICIPATE)) \bowtie < P.CON_ID=J.CON_ID>\sigma$

P(J,σ JOB_ITLE='Database Manager' (JOB_INFORMATION))

Q34) LIST THE TOP 3 DEVELOPERS WHO GOT MAXIMUM CERTIFICATES IN 2020

SQL Queries

SELECT D_NAME FROM DEVELOPER AS D JOIN

(SELECT CON_ID,DEV_ID,COUNT(CON_ID) AS C FROM CERTIFICATE GROUP BY CON_ID,DEV_ID ORDER BY DEV_ID DESC LIMIT 3)

AS C1 ON D.DEV_ID=C1.DEV_ID JOIN

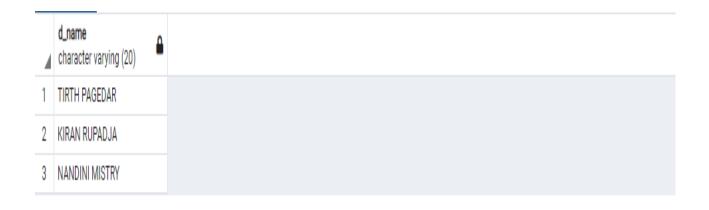
(SELECT EXTRACT(YEAR FROM CREATE_AT) AS YEAR, CON_ID FROM CONTEST)

AS CON1 ON C1.CON_ID=CON1.CON_ID WHERE CON1.YEAR=2020

• Relational Algebra

 Π D_NAME(DEVELOPER) ⋈<D1.DEV_ID=C1.DEV_ID> σ CON_ID,DEV_ID(
(σ < LIMIT 3 > \mathcal{F} <COUNT(CON_ID)(CERTIFICATE) >)⋈<C1.CON_ID=CON1.CON_ID> (σ < YEAR=2020 > P(YEAR, \mathcal{F} <EXTRACT(YEAR FROM CREATE AT)) > (CONTEST))

Output



Q35) FIND THE NUMBER OF 'A GRADE' COMPANY WHO HAD CREATED THE CONTEST.

SQL Queries

SELECT COUNT(COM_ID) FROM COMPANY WHERE COMPANY_SIZE='GRADE A' AND COM_ID IN(SELECT COM_ID FROM CONTEST)

• Relational Algebra

σ COMPANY_SIZE= 'GRADE A' (%<COUNT(CON_ID)(COMPANY)>)

SEMI-JOIN CONTEST

	int 🖣
1	2

.....

Q36) FIND THE NAME OF DEVELOPERS WHO HAS EITHER SKILL ID SKL0001(C++) OR SKL0003(JAVA)

SQL Queries

SELECT DEV_ID,D_NAME FROM DEVELOPER WHERE DEV_ID IN (SELECT DEV_ID FROM CONTAINS WHERE SKILL_ID='SKI0001' UNION SELECT DEV_ID FROM CONTAINS WHERE SKILL_ID='SKI0003')

• Relational Algebra

 σ D_NAME(DEVELOPER) SEMI-JOIN

σ SKILL ID= 'C++'(CONTAINS) U

σ SKILL_ID= 'JAVA' (CONTAINS)

• Output



Q37) FIND THE NAME OF DEVELOPER WHO HAS ATTENDED HIGHEST NUMBER OF ASSIGNMENT

SQL Queries

SELECT D_NAME FROM DEVELOPER AS D JOIN

(SELECT DEV_ID,COUNT(ASSI_ID) AS NUM FROM PRACTICE_ASSIGNMENT GROUP BY DEV_ID ORDER BY NUM DESC LIMIT 1)

AS C1 ON D.DEV_ID=C1.DEV_ID

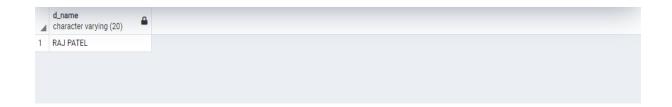
Relational Algebra

R1 -> σ D NAME(DEVELOPER)

R2 -> σ <LIMIT 1> P(NUM, %<NUM, COUNT(ASSI_ID)>(PRACTICE_ASSIGNMENT))

R1 ⋈<DEVELOPER.DEV_ID= PRACTICE_ASSIGNMENT.DEV_ID>

• Output



.....

Q38) FIND THE NAME OF IT COMAPNY WHO HAS CREATED AT LEAST 3 CONTESTS.

SQL Queries

SELECT C_NAME, COMPANY_TYPE FROM COMPANY AS C JOIN

(SELECT COM_ID,COUNT(CON_ID) AS NUM FROM CONTEST AS CON GROUP BY CON_ID) AS CON ON C.COM_ID=CON.COM_ID

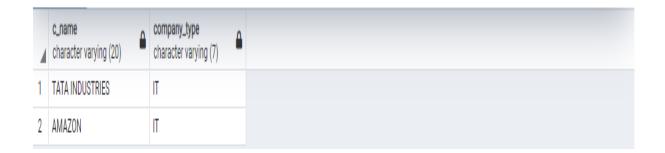
WHERE CON.NUM=1 AND C.COMPANY_TYPE='IT'

• Relational Algebra

 σ COMPANY_TYPE= 'IT' (COMPANY) \bowtie < COMPANY.COM_ID = CONTEST.COM_ID>

 σ NUM>=3 \Re <COUNT(COM_ID) -> NUM > (CONTEST)

• Output



Q39) LIST OUT ALL THE DEVELOPERS WHOSE SCORE IS ABOVE 1500.

• SQL Queries

SELECT * FROM DEVELOPER NATURAL JOIN PROFILE WHERE SCORE > 1500;

Relational Algebra

 $\sigma_{SCORE>1500}(DEVELOPER \bowtie PROFILE)$

Output

Data Output Explain Messages Notifications

1	dev_id character (7)	create_at timestamp with time zone	d_name character varying (20)	d_email character varying (50)	d_contact_info numeric (10)	d_address character varying (100) □	d_country character varying (20) □	gender character (1)
1	DEV0001	2020-10-09 14:31:12.830583+05:30	RAJ PATEL	rajpatel@gmail.com	9265999123	7741 Bellevue Drive Auburndale,	INDIA	M
2	DEV0002	2020-10-09 14:31:12.830583+05:30	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee, F	INDIA	М
3	DEV0003	2020-10-09 14:31:12.830583+05:30	KIRAN RUPADJA	kiran@gmail.com	9265999345	8399 Bradford Drive Hartselle, A	USA	F
4	DEV0004	2020-10-09 14:31:12.830583+05:30	TIRTH PAGEDAR	tirth@gmail.com	9265999456	77 Spring Street Rockaway, NJ 0	USA	М

Q40) LIST OUT ALL THE DEVELOPERS WHO HAS MORE THAN FIVE CERTIFICATES AND STILL NOT RECRUITED BY ANY COMPANY

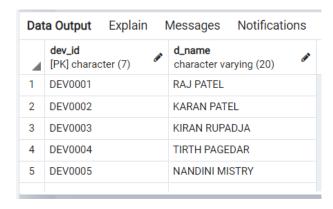
SQL Queries

Select Developer.* from Developer as d join certificate as c on d.Dev_id =c.Dev_id GROUP BY d.DEV_ID having count(c.Dev_id) >5;

• Relational Algebra

σ_{COUNT(DEV_ID)}>5(DEV_ID ℑ_{DEVELOPER.*}(DEVELOPER ⋈ CERTIDICATE))

Output



Q41) FIND OUT WHICH COUNTRY HAS HIGHEST DEVELOPERS ON OUR PLATFORM.

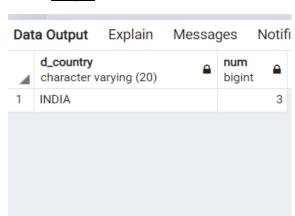
SQL Queries

SELECT D_COUNTRY,COUNT(D_COUNTRY) AS NUM FROM DEVELOPER GROUP BY D_COUNTRY ORDER BY NUM DESC LIMIT 1;

• Relational Algebra

 σ order by num desc and limit 1(D_country3D_country,count(D_country)->num(D_COUNTRY))

• Output



Q42) FIND DEVELOPERS OF A PARTICULAR COUNTRY

• SQL Queries

Select * from Developer where d_country = 'India'

• Relational Algebra

 $\sigma_{\text{D_COUNTRY='INDIA'}}(\text{DEVELOPER})$

Output

4	dev_id [PK] character (7)	d_name character varying (20)	d_email character varying (50)	d_contact_info numeric (10)	d_address character varying (100)	d_country character varying (20)	gender character (1)	dob date
1	DEV0001	RAJ PATEL	rajpatel@gmail.com	9265999123	7741 Bellevue Drive Auburnda	INDIA	М	1999-12
2	DEV0002	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee	INDIA	М	1999-12.
3	DEV0005	NANDINI MISTRY	nandini@gmail.com	9265999567	9 South Poor House Road Lut	INDIA	F	1999-07.

Q43) LIST QUESID OF TOP 3 MOST COMFORTABLE(MOST SOLVED) QUESTIONS BASE ON SKILL ID SK0001.

SQL Queries

SELECT QUE_ID,COUNT(QUE_ID) AS COUNT,SKILL_ID FROM QUESTION NATURAL JOIN (SELECT QUE_ID from ATTEND_QUE_ASSIGNMENT WHERE STATUS='SOLVED' UNION ALL SELECT QUE_ID FROM ATTEND_QUE_CONTEST WHERE STATUS='COMPLETE') AS R1 WHERE SKILL_ID='SKI0001' GROUP BY QUE_ID ORDER BY COUNT DESC LIMIT 3;

• Relational Algebra

 σ skill_id='ski0001' and limit 3 and order by count(\Im que_id,count(que_id)->count,skill_id (QUESTION \bowtie ρ-(R1, π que_id(σ status='solved'(ATTEND_QUE_ASSIGNMENT)) U π que_id(σ status='complete'(ATTEND_QUE_CONTEST)))

• Output

Dat	a Output	Explain	ı	Messag	es	Notifications	6
4	que_id [PK] charac	oter (7)		count bigint		skill_id character (7)	S
1	QUE0002				1	SKI0001	

Q44) LIST TOP 10 DIFFICULT TEST CASES (BASED ON PEOPLE'S ATTEMPT AND UNABLE TO SOLVE IT).

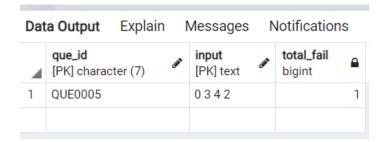
• SQL Queries

SELECT QUE_ID,INPUT,COUNT(*) AS TOTAL_FAIL FROM TESTCASE NATURAL JOIN (SELECT * from attend_testcase_contest where status='FAIL' UNION ALL SELECT * from attend_testcase_assignment where status='FAIL') AS ALL_FAIL GROUP BY QUE_ID,INPUT ORDER BY TOTAL_FAIL LIMIT 10;

Relational Algebra

 $\sigma_{\text{limit 10 AND ORDER BY TOTAL}}$ (QUE_ID,INPUT\$QUE_ID,INPUT,COUNT(*)->TOTAL(TESTCASE \bowtie ρ(ALL_FAIL, $\sigma_{\text{status}='\text{fail'}}$ (ATTENT_TESTCASE_CONTEST)) U $\sigma_{\text{status}='\text{fail'}}$ (attend_testcase_assignment)))

• Output



Q45) LIST QUESID OF TOP 3 MOST COMFORTABLE(MOST SOLVED) QUESTIONS BASE ON SKILL ID SK0001.

• SQL Queries

SELECT QUE_ID,COUNT(QUE_ID) AS COUNT,SKILL_ID FROM QUESTION NATURAL JOIN (SELECT QUE_ID from ATTEND_QUE_ASSIGNMENT WHERE STATUS='SOLVED' UNION ALL SELECT QUE_ID FROM ATTEND_QUE_CONTEST WHERE STATUS='COMPLETE') AS R1 WHERE SKILL ID='SKI0001' GROUP BY QUE ID ORDER BY COUNT DESC LIMIT 3;

• Relational Algebra

$$\begin{split} &\sigma_{\text{SKILL_ID}=\text{'SKI0001'} \text{ and order by count and limit 3} \big(\text{que_id} \mathfrak{I}_{\text{Que_id},\text{count}}(\text{que_id})\text{->count,skill_id from} \\ &\left(\text{QUESTION} \bowtie \rho(\text{R1},\pi_{\text{QUe_iD}}(\sigma_{\text{STATUS}=\text{'SOLVED'}}(\text{ATTEND_QUE_ASSIGNMENT})) \text{ U} \\ &\pi_{\text{QUe_id}}(\sigma_{\text{STATUS}=\text{'COMPLETE'}}(\text{ATTEND_QUE_CONTEST})) \text{)))} \end{split}$$

Dat	a Output	Explain	Messages		es	Notifications	
4	que_id [PK] character (7)		S	count bigint	<u></u>	skill_id character (7)	
1	QUE0002				1	SKI0001	

.....

Q46) LIST OUT ALL THE JAVA DEVELOPERS WHO GOT SELECTED IN PARTICULAR COMPANY FROM 2015 TO 2020.

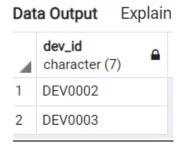
SQL Queries

SELECT DEV_ID FROM participate NATURAL JOIN PROFILE NATURAL JOIN SKILL WHERE S_NAME='JAVA' AND IS RECRUITED='T' AND participate.CREATE AT between '2018-01-01 00:00:00' AND '2022-01-01 00:00:00';

• Relational Algebra

 $\Pi_{S_DEV_ID}(\sigma_{S_NAME='JAVA'})$ and is_recruited='t' and participate.created_at between 2018-01-01 00:00:00 and 2022-01-01 00:00:00 (PARTICIPATE \bowtie SKILL))

• Output



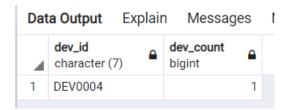
Q47) RETRIEVE DEVELOPERID, WHO HAS ANSWERED MAXIMUM NUMBER OF QUESTIONS IN PYTHON.

SQL Queries

SELECT DEV_ID,COUNT(DEV_ID) AS DEV_COUNT FROM CONTAINS NATURAL JOIN(SELECT DEV_ID FROM attend_que_assignment UNION ALL SELECT DEV_ID FROM attend_que_contest) AS ALL_ATTENT WHERE SKILL_ID='SKI0004' GROUP BY DEV_ID ORDER BY DEV_COUNT DESC;

• Relational Algebra

$$\begin{split} &\sigma_{\text{SKILL_ID='SKI0004'}} \text{ and order by dev_count desc} (\mathfrak{T}_{\text{DEV_ID,COUNT(DEV_ID})-} \\ &>_{\text{DEV_COUNT}} (\text{COUNTAINS} \bowtie \rho(\text{ALL_ATTEND}, \pi_{\text{DEV_ID}}(\text{ATTEND_QUE_ASSIGNMENT}) \ U \\ &\pi_{\text{DEV_ID}}(\text{ATTEND_QUE_CONTEST})))) \end{split}$$



Q48) LIST THE NAME OF THE SKILL WHICH, HAVE MAXIMUM NUMBER OF DEVELOPERS ASSIGN.

SQL Queries

SELECT SKILL_ID, COUNT(SKILL_ID) AS SKILL FROM SKILL NATURAL JOIN CONTAINS GROUP BY (SKILL_ID) ORDER BY SKILL DESC LIMIT 1;

• Relational Algebra

σ order by skill desc and limit 1 (skill_id, skill_id, count(skill_id)->skill (SKILL ⋈ CONTAINS))

• Output



.....

Q49) FIND NAME OF COMPANY WHICH HAS LEATS NUMBER OF INTAKE.

• SQL Queries

SELECT company.c_name from job_information join contest on(job_information.con_id=contest.con_id) join company on(contest.com_id=company.com_id) where toatal_intake=(select min(toatal_intake) from job_information);

Relational Algebra

R1: $\mathfrak{I}_{min(toatal_intake)}(job_information)$

 $\pi_{company.c_name}(\sigma_{toatal_intake=R1}(job_information \bowtie_{job_information.con_id=contest.con_id} contest \bowtie_{contest.com_id=company.com_id} company)$



Q50) FIND MAX NUMBER OF QUESTION SOLVE BASED ON JAVA BY INDIVIDUAL DEVELOPER INCLUDING ALL CONTEST;

• SQL Queries

select max(max_que) from (select dev_id,count(question.que_id) as max_que from attend_que_contest join question on (attend_que_contest.que_id=question.que_id) where skill_id='SKI0003' and status='COMPLETE' group by dev_id) as r1

Relational Algebra

$$\label{eq:contest_que_id} \begin{split} &\mathfrak{Z}_{max(max_que)}(\ \rho(r1\ ,\ \textbf{dev}_\mathsf{id} \mathfrak{Z}_{dev_\mathsf{id},count(question.que_\mathsf{id})->max_que}(\ \sigma_{status='complete'\ and\ skill_\mathsf{id}='SKI003'}\\ &(attend_que_contest\bowtie_{attend_que_contest.que_\mathsf{id}=question.que_\mathsf{id}}(\ \sigma_{status='complete'\ and\ skill_\mathsf{id}='SKI003'})\)\) \end{split}$$

• Output

Data Output								
4	maxque integer	<u></u>						
1		10	·					