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

$$\sigma(\text{DEVELOPER}) \bowtie P(\text{NONEPARTICIPANT},\Pi_{\text{DEVELOPER.DEV\_ID}}(\sigma_{\text{PARTICIPATE.DEV\_ID IS NULL}}(\text{DEVELOPER} \bowtie_{\text{DEVELOPER.DEV\_ID=PARTICIPATE.DEV\_ID}} \text{PARTICIPATE})) \cup \Pi_{\text{DEVELOPER.DEV\_ID}}(\sigma_{\text{PRACTICE\_ASSIGMENT.DEV\_ID IS NULL}}(\text{DEVELOPER} \bowtie_{\text{DEVELOPER.DEV\_ID=PRACTICE\_ASSIGMENT.DEV\_ID}} \text{PRACTICE\_ASSIGNMENT})))$$

OUTPUT

Data Output | Explain | Messages | Notifications

	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	M	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

$$\sigma(\text{DEVELOPER}) \bowtie P(\text{ONLYCONTEST}, \Pi_{\text{DEVELOPER.DEV\_ID}}(\sigma_{\text{PARTICIPATE.DEV\_ID IS NULL}}(\text{DEVELOPER} \bowtie_{\text{DEVELOPER.DEV\_ID=PARTICIPATE.DEV\_ID}} \text{PARTICIPATE})) - \Pi_{\text{DEVELOPER.DEV\_ID}}(\sigma_{\text{PRACTICE\_ASSIGMENT.DEV\_ID IS NULL}}(\text{DEVELOPER} \bowtie_{\text{DEVELOPER.DEV\_ID=PRACTICE\_ASSIGMENT.DEV\_ID}} \text{PRACTICE\_ASSIGNMENT})))$$

OUTPUT

Data Output | Explain | Messages | Notifications

	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	DEV0004	TIRTH PAGEDAR	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

$$\sigma_{\text{LIMIT 1}}(\text{QUESTION} \bowtie P(\text{TCASE}, {}_{\text{QUE\_ID}}\Im_{\text{QUE\_ID,COUNT\_ID}}(\text{TESTCASE}) \text{ ORDER BY QUE\_ID DESC }))$$

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

OUTPUT

Data Output | Explain | Messages | Notifications

	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$\bowtie$CONTAINS$\bowtie$SKILL$\bowtie$PARTICIPATE P (CONTESTBYBIRLA, Π_{DEV_ID} (σ _NAME='JAVA' AND AGE(DOB)>'20 YEAR': INTREVAL(COMPANY$\bowtie_{COMPANY.COM_ID=CONTEST.COM_ID}$CONTEST$\bowtie_{CONTEST.CON_ID=PARTICIPATE.CONID}$PARTICIPATE))))

OUTPUT

Data Output Explain Messages Notifications

	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

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$\bowtie_{ASSIGNMENT.ADM_ID=ADMIN.ADM_ID}$ADMIN)

OUTPUT

Data Output Explain Messages Notifications

	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 time	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 $\bowtie$P(WINERLIST,Π_{DEV_ID}(σ C_NAME='COMPETETIVE TEST BY AMAZON' AND IS_RECRUIRED='T'(CONTEST $\bowtie_{CONTEST.CON_ID=PARTICIPATE.CON_ID}$PARTICIPATE))))

OUTPUT

Data Output Explain Messages Notifications

	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

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

$\Pi_{JOB\_TITLE,DEVELOPER.*}$(σ IS_RECRUITED='T'(DEVELOPER $\bowtie$PARTICIPATE$\bowtie$JOB_INFORMATION))

OUTPUT

Data Output Explain Messages Notifications

	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

σ (DEVELOPER$\bowtie$P(ATLEAST2, σ IS_RECRUITED='T' AND COUNT(*)>1($_{DEV\_ID}\Im_{DEV\_ID}$(PARTICIPATE))))

OUTPUT

Data Output Explain Messages Notifications

	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	M	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:= $\Im_{MAX(SALARY)->MAX}$(JOB_INFORMATION)

σ IS_RECUIRTED='T' AND SALARY=R1.MAX
(DEVELOPER$\bowtie$PARTICIPATE$\bowtie$JOB_INFORMATION)

OUTPUT

Data Output Explain Messages Notifications

	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	M

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, $_{SKILL\_ID}\Im_{SKILL\_ID}$ ORDER BY SKILLID DESC LIMIT 1)

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

OUTPUT

Data Output | Explain | Messages | Notifications

	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

Π_{DEV_ID}(PARITICIPATE) –Π_{DEV_ID}(P(R1,$\Pi_{CONTEST.CON\_ID,PARTICIPATE.DEV\_ID}$(σ $_{COMPANY.C\_NAME='AMAZON'}$(PARTICIPATE X CONTEST⋈$_{CONTEST.COM_ID=COMPANY.COM_ID}$COMPANY) – Π_{CON_ID,DEV_ID}(PARTICIPATE))))

OUTPUT

Data Output | Explain | Me

	dev_id character (7)
1	DEV0003

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

Data Output | Explain

	dev_id character (7)
1	DEV0020

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

Data Output | Explain | Mes

	dev_id character (7)
1	DEV0005

Q14) 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

Data Output

	dev_id character varying	num integer
1	DEV0001	10
2	DEV0002	9
3	DEV0003	8
4	DEV0004	6
5	DEV0005	5
6	DEV0025	2
7	DEV0006	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= $\sigma_{\text{IS\_RECRUITED='T'}}(\Im_{\text{AVG(SALARY)-> AVG}}(\text{JOB\_INFORMATION} \bowtie \text{PAERTICIPATE}))$

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

OUTPUT

Data Output Explain Messages Notifications

	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	M	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

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

OUTPUT

Data Output Explain Messages Notifications

	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

ΠD_NAME(ρ(D,DEVELOPER)⋈D.DEV_ID=C1.DEV_ID ρ(C1, σ ORDER BY NUM DESC AND LIMIT 1 (DEV_IDℑDEV_ID,COUNT(ASSI_ID)(PRACTICE_ASSIGNMENT)))'

OUTPUT

Data Output Explain Messages

	d_name character varying (20)
1	NANDINI MISTRY

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

$\sigma_{\text{LIMIT1 AND ORDER BY QUE\_ID}}(_{\text{QUE\_ID}}\Im_{\text{QUE\_ID,COUNT(QUE\_ID)}}(P(R1,$

$\Pi_{\text{QUE\_ID}}(\Sigma_{\text{STATUS='UNSOLVED'}}(\text{ATTEND\_QUE\_ASSIGNMENT}))$

U

$\Pi_{\text{QUE\_ID}}(\sigma_{\text{STATUS='UNCOMPLETE'}}(\text{ATTEND\_QUE\_CONTEST})))))$

OUTPUT

Data Output | Explain | Messag

	que_id character (7)	count bigint
1	QUE0005	1

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

$\sigma_{\text{COUNT(DEV\_ID)>=1 AND ORDER BY DEV\_ID}}(_{\text{DEV\_ID}}\Im_{\text{DEV\_ID,COUNT(DEV\_ID)}}(\text{CERTIFICATE}))$

OUTPUT

Data Output

	dev_id character varying	count integer
1	DEV0001	10
2	DEV0016	8
3	DEV0003	7
4	DEV0004	6
5	DEV0002	5
6	DEV0005	4
7	DEV0006	3
8	DEV0025	2
9	DEV0024	2
10	DEV0014	1

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

$\sigma_{\text{IS\_RECRUITED='T' AND ORDER BY COM\_ID AND LIMIT 1}}(_{\text{COM\_ID}}\Im_{\text{COM\_ID-}}$
(PARTICIPATE$\bowtie_{\text{PARTICIPATE.CON_ID=CONTEST.CON_ID}}$CONTEST))

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

OUTPUT

Data Output Explair

	com_id character (7)
1	COM0003

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

$\pi_{\text{d\_name}}(\sigma_{\text{CURRENT\_EDUCATION='MSCIT'}}(\text{DEVELOPER}\bowtie\text{PROFILE}))$

OUTPUT

Data Output Explain Messa

	d_name character varying (20)
1	JAGDISH PATEL
2	ROHIT TANTI
3	AESHA KAYESTHA
4	PRACHI DESAI

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

$\Pi_{D\_NAME}(\sigma_{OUTPUT='206'}(ATTEND\_TESTCASE\_ASSIGNMENT \bowtie TESTCASE))$

OUTPUT

Data Output | Explain | Messa

	d_name character varying (20)
1	RAJ PATEL
2	NANDINI MISTRY

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

σ LIMIT 1 AND ORDER BY $DEV\_ID(DEV\_ID\Im_{DEV\_ID,COUNT(DEV\_ID)}(DEVELOPER \bowtie ATTEND\_QUE\_ASSIGNMENT))$

OUTPUT

Data Output | Explain | Messages

	dev_id [PK] character (7)	count bigint
1	DEV0001	3

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

σ LIMIT 1 AND ORDER BY MAX(TOTAL_SCORE) DESC
$(_{TOTAL\_SCORE,D\_NAME}\Im_{D\_NAME,MAX(TOTAL\_SCORE)}(DEVELOPER \bowtie CONTONS \bowtie QUESTION))$

OUTPUT

Data Output | Explain | Messages | Notific

	d_name character varying (20)	max integer
1	TIRTH PAGEDAR	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

$\pi_{D\_NAME,DIFFICULTY\_LEVEL}(\ \sigma_{DIFFICULTY\_LEVEL='MODERATE' \text{ AND } TITLE='INHERITED\ CODE'}(DEVELOPER \bowtie CONTAINS \bowtie QUESTION))$

OUTPUT

Data Output | Explain | Messages | Notifications

	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

$\sigma_{\text{LIMIT 5 AND D\_COUNTRY='INDIA' AND ORDER BY DESC}}$
$(\text{D\_NAME,D\_COUNTRY,TOATAL\_SCORE}\Im_{\text{D\_NAME,MAX(TOTAL\_SCORE)}})$

OUTPUT

Data Output | Explain | Messages | Notifica

	d_name character varying (20)	max integer
1	JITARTH shah	2678
2	PRACHI DESAI	2624
3	HITDHARM DOSHI	2623
4	RAJ PATEL	2565
5	JAGDISH PATEL	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)

- **Output**

Data Output | Explain | Messages | Notifications

	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 Auburndale,...	INDIA	M	1999-12-20
2	DEV0005	NANDINI MISTRY	nandini@gmail.com	9265999567	9 South Poor House Road Luthe...	INDIA	F	1999-07-23

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 ⋈ PARTICIPATE ⋈ CONTEST ⋈ COMPANY)

- **Output**

Data Output Explain Messages Notifications

	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	M	1999-12-05

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**

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

- **Output**

Data Output Explain Messages Notifications

	dev_id [PK] character (7)	con_id [PK] character (10)	score integer	is_recruited boolean
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**

$R1=\Im_{MAX(END\_TIME\_DATE - START\_TIME\_DATE)->MAX}$

$\sigma_{END\_TIME-START\_TIME\_DATE=R1.MAX}(CONTEST)$

- **Output**

Data Output | Explain | Messages | Notifications

	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**

$\prod P(D1,\sigma D\_name(Developer)) \bowtie <P1.Dev\_ID=D1.Dev\_ID>$

$P(P1,\sigma\ Dev\_ID, Qualification='BCA'(profile) \bowtie <D1.Dev\_ID=P2.Dev\_ID>$

$P(P2,\sigma\ Dev\_ID, is\_recruited='T'(participate))\sigma$

- **Output**

	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**

∏P(C, σ C_name,Company_type=’non-IT’(Company))

⋈<C.COM_ID=CON1.COM_ID >

P(CON1,σ CON_ID,COM_ID (Contest)⋈< J.CON_ID=CON1.CON_ID>

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

- **Output**

	c_name character varying (20)	company_type character varying (7)
1	BIRLA INDUSTRIES	NON-IT

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**

∏P(D, σ D_name(DEVELOPER))

⋈<D.DEV_ID=C1.DEV_ID >

P(C1,σ SKILL_ID=’SKIOO1’ OR SKILL-_ID=’SKI002’ (Contest)⋈< P.DEV_ID=C1.DEV_ID>

P(P,σ DEV_ID,CON_ID,IS_RECRUITED=’T’(PARTICIPATE)) ⋈<P.CON_ID=J.CON_ID>σ

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

- **Output**

	d_name character varying (20)
1	KARAN PATEL

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 > ℱ<COUNT(CON_ID)(CERTIFICATE) >)⋈<C1.CON_ID=CON1.CON_ID> (σ< YEAR=2020 > P(YEAR,ℱ<EXTRACT(YEAR FROM CREATE_AT)) >(CONTEST))

- **Output**

	d_name character varying (20)
1	TIRTH PAGEDAR
2	KIRAN RUPADJA
3	NANDINI MISTRY

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’ (ℱ<COUNT(CON_ID)(COMPANY)>)

SEMI-JOIN CONTEST

- **Output**

	count bigint
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**

	dev_id [PK] character (7)	d_name character varying (20)
1	DEV0001	RAJ PATEL
2	DEV0002	KARAN PATEL
3	DEV0003	KIRAN RUPADJA
4	DEV0005	NANDINI MISTRY

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,ℱ<NUM,COUNT(ASSI_ID)>(PRACTICE_ASSIGNMENT))

R1 ⋈<DEVELOPER.DEV_ID= PRACTICE_ASSIGNMENT.DEV_ID>

- **Output**

	d_name character varying (20)
1	RAJ PATEL

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) ⋈<COMPANY.COM_ID = CONTEST.COM_ID>

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

- **Output**

	c_name character varying (20)	company_type character varying (7)
1	TATA INDUSTRIES	IT
2	AMAZON	IT

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

	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	M
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	M

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**

$$\sigma_{COUNT(DEV\_ID)>5}(_{DEV\_ID}\Im_{DEVELOPER.*}(DEVELOPER \bowtie CERTIDICATE))$$

- **Output**

Data Output | Explain | Messages | Notifications

	dev_id [PK] character (7)	d_name character varying (20)
1	DEV0001	RAJ PATEL
2	DEV0002	KARAN PATEL
3	DEV0003	KIRAN RUPADJA
4	DEV0004	TIRTH PAGEDAR
5	DEV0005	NANDINI MISTRY

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**

$$\sigma_{\text{ORDER BY NUM DESC AND LIMIT 1}}(_{D\_COUNTRY}\Im_{D\_COUNTRY,COUNT(D\_COUNTRY)->NUM}(D\_COUNTRY))$$

- **Output**

Data Output | Explain | Messages | Notifi

	d_country character varying (20)	num bigint
1	INDIA	3

Q42) FIND DEVELOPERS OF A PARTICULAR COUNTRY

- **SQL Queries**

Select * from Developer where d_country = 'India'

- **Relational Algebra**

$\sigma_{D\_COUNTRY='INDIA'}(DEVELOPER)$

- **Output**

Data Output | Explain | Messages | Notifications

	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	M	1999-12...
2	DEV0002	KARAN PATEL	karan@gmail.com	9265999234	356 S. Lookout St. Kissimmee...	INDIA	M	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**

$\sigma_{SKILL\_ID='SKI0001' \text{ AND LIMIT 3 AND ORDER BY } COUNT}(\Im_{QUE\_ID,COUNT(QUE\_ID)->COUNT,,SKILL\_ID}(QUESTION \bowtie \rho_{(R1,\pi_{QUE\_ID}(\sigma_{STATUS='SOLVED'}(ATTEND\_QUE\_ASSIGNMENT))} \cup \pi_{QUE\_ID}(\sigma_{STATUS='COMPLETE'}(ATTEND\_QUE\_CONTEST))))$

- **Output**

Data Output | Explain | Messages | Notifications

	que_id [PK] character (7)	count bigint	skill_id character (7)
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**

$\sigma_{\text{limit 10 AND ORDER BY TOTAL}}(_{\text{QUE\_ID,INPUT}}\Im_{\text{QUE\_ID,INPUT,COUNT(*)->TOTAL}}(\text{TESTCASE} \bowtie \rho(\text{ALL\_FAIL},\sigma_{\text{status='fail'}}(\text{ATTENT\_TESTCASE\_CONTEST})) \cup \sigma_{\text{status='fail'}}(\text{attend\_testcase\_assignment})))$

- **Output**

Data Output | Explain | Messages | Notifications

	que_id [PK] character (7)	input [PK] text	total_fail bigint
1	QUE0005	0 3 4 2	1

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**

$\sigma_{\text{SKILL\_ID='SKI0001' AND ORDER BY COUNT AND LIMIT 3}}(_{\text{QUE\_ID}}\Im_{\text{QUE\_ID,COUNT(QUE\_ID)->COUNT,SKILL\_ID FROM}}(\text{QUESTION} \bowtie \rho(\text{R1},\pi_{\text{QUE\_ID}}(\sigma_{\text{STATUS='SOLVED'}}(\text{ATTEND\_QUE\_ASSIGNMENT})) \cup \pi_{\text{QUE\_ID}}(\sigma_{\text{STATUS='COMPLETE'}}(\text{ATTEND\_QUE\_CONTEST})))))$

- **Output**

Data Output | Explain | Messages | Notifications

	que_id [PK] character (7)	count bigint	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**

$\Pi_{S\_DEV\_ID}(\sigma_{S\_NAME='JAVA' \text{ AND } IS\_RECRUITED='T' \text{ AND } PARTICIPATE.CREATED\_AT \text{ BETWEEN } 2018\text{-}01\text{-}01\ 00:00:00 \text{ AND } 2022\text{-}01\text{-}01\ 00:00:00}(PARTICIPATE \bowtie SKILL))$

- **Output**

Data Output Explain

	dev_id character (7)
1	DEV0002
2	DEV0003

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**

$\sigma_{SKILL\_ID='SKI0004' \text{ AND ORDER BY } DEV\_COUNT \text{ DESC}}(\Im_{DEV\_ID,COUNT(DEV\_ID)->DEV\_COUNT}(COUNTAINS \bowtie \rho(ALL\_ATTEND, \pi_{DEV\_ID}(ATTEND\_QUE\_ASSIGNMENT) \cup \pi_{DEV\_ID}(ATTEND\_QUE\_CONTEST))))$

- **Output**

Data Output Explain Messages

	dev_id character (7)	dev_count bigint
1	DEV0004	1

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$\Im$SKILL_ID,COUNT(SKILL_ID)->SKILL (SKILL$\bowtie$CONTAINS))

- **Output**

Data Output | Explain | Messages

	skill_id [PK] character (7)	skill bigint
1	SKI0001	3

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: $\Im_{min(toatal\_intake)}$(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))

- **Output**

Data Output

	c_name character varying (20)
1	BIRLA INDUSTRIES

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**

$\Im_{max(max\_que)}(\rho(r1 , \textbf{dev}\_id\Im_{dev\_id,count(question.que\_id)->max\_que}(\sigma_{status='complete' \text{ and } skill\_id='SKI003'} (attend\_que\_contest \bowtie_{attend\_que\_contest.que\_id=question.que\_id} question))))$

- **Output**

Data Output

	maxque integer
1	10
