

DBMS ASSIGNMENT-5

SUBMITTED BY:

Vytla Harsha Vardhan	18bcs118
Pokala Dattatreya	18bcs067
Rama Dundi Saketh	18bcs076
Shaik Fharook	18bcs091
Y Mokshith Ramendra	18bcs112
Umesh Anand Babu	18bcs105
Perumala Tushar	18bcs065

A Database named ass4 is created and created tables named registration,festday1 and festday2 which contains information regarding students registered for fest,students attended io fest on day1 and on day2 respectively

Registration table:

#	stuid	stname	clgid	gen	mobnum	addr	regid	
1	1	ram	liitdwd	male	9496675656	hubli	r01	
2	2	priya	liitdwd	female	9396675656	hubli	r02	
3	3	sudha	sdm	female	9496665656	dwd	r03	
4	4	divi	kle	female	9496685656	hubli	r04	
5	5	abhi	kle	male	9496675556	hubli	r05	
6	6	sam	sdm	male	9496675654	dwd	r06	
7	7	rita	liitdwd	female	9496675657	hubli	r07	
8	8	monal	sdm	female	9296675657	dwd	r08	
9	9	tom	jain	male	8496675657	hubli	r09	

Festday1 table:

#	regid	clgid	stname	gen	entry	idproof	
1	r01	liitdwd	ram	male	09:00:00	aadhar	
2	r02	liitdwd	priya	female	11:00:00	clgid	
3	r03	sdm	sudha	female	09:00:00	voterid	
4	r04	kle	divi	female	10:00:00	aadhar	
5	r08	sdm	monal	female	10:00:00	aadhar	
6	r10	sdm	jams	male	11:00:00	aadhar	
7	r11	kle	uma	female	11:00:00	aadhar	
8	r12	kle	joss	male	11:00:00	clgid	
9	r13	liitdwd	stokes	male	12:00:00	aadhar	
10	r14	jain	sita	female	12:00:00	aadhar	

Festday2 table:

#	regid	clgid	stname	gen	entry	idproof
1	r01	iltdwd	ram	male	09:00:00	aadhar
2	r04	kle	divi	female	09:00:00	aadhar
3	r05	kle	male	abhi	09:00:00	voterid
4	r07	iltdwd	rita	female	10:00:00	clgid
5	r08	sdm	monal	female	09:00:00	aadhar
6	r09	jain	tom	male	10:00:00	clgid
7	r11	kle	uma	female	11:00:00	aadhar
8	r13	iltdwd	stokes	male	12:00:00	aadhar
9	r14	sdm	sita	female	11:00:00	aadhar
10	r15	jain	avi	male	12:00:00	clgid

IMPLEMENTING JOIN:

Finding students who have registered and came to fest on day1 using join

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
81
82
83 select r.regid, r.stname, r.clgid, f1.entry
84 from registration as r
85 join festday1 as f1
86 on r.regid=f1.regid;
87
```

The Result Grid shows the output of the query:

#	regid	stname	clgid	entry
1	r01	ram	iltdwd	09:00:00
2	r02	priya	iltdwd	11:00:00
3	r03	sudha	sdm	09:00:00
4	r04	divi	kle	10:00:00
5	r08	monal	sdm	10:00:00

The Action Output pane at the bottom shows the query execution details:

#	Time	Action	Message	Duration / Fetch
265	13:45:04	select r.regid, r.stname, r.clgid, f1.entry from registration as r j...	5 row(s) returned	0.00044 sec / 0.000...

Query Completed

Finding students who have registered and came to fest on day2 using join

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
87
88 select r.regid, r.stname, r.clgid, f2.entry
89 from registration as r
90 join festday2 as f2
91 on r.regid=f2.regid;
92
93
```

The Result Grid displays the following data:

#	regid	stname	clgid	entry
1	r01	ram	lltdwd	09:00:00
2	r04	divi	kle	09:00:00
3	r05	abhl	kle	09:00:00
4	r07	rita	lltdwd	10:00:00
5	r08	monal	sdm	09:00:00
6	r09	tom	jain	10:00:00

The Action Output pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
266	13:46:04	select r.regid, r.stname, r.clgid, f2.entry from registration as r j...	6 row(s) returned	0.00052 sec / 0.000...

Query Completed

Finding students who have come to fest on both the days wheather registered or not

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
92
93
94 select f1.regid, f1.stname, f1.entry, f2.entry
95 from festday1 as f1
96 join festday2 as f2
97 on f1.regid=f2.regid;
98
```

The Result Grid displays the following data:

#	regid	stname	entry	entry
1	r01	ram	09:00:00	09:00:00
2	r04	divi	10:00:00	09:00:00
3	r08	monal	10:00:00	09:00:00
4	r11	uma	11:00:00	11:00:00
5	r13	stokes	12:00:00	12:00:00
6	r14	sita	12:00:00	11:00:00

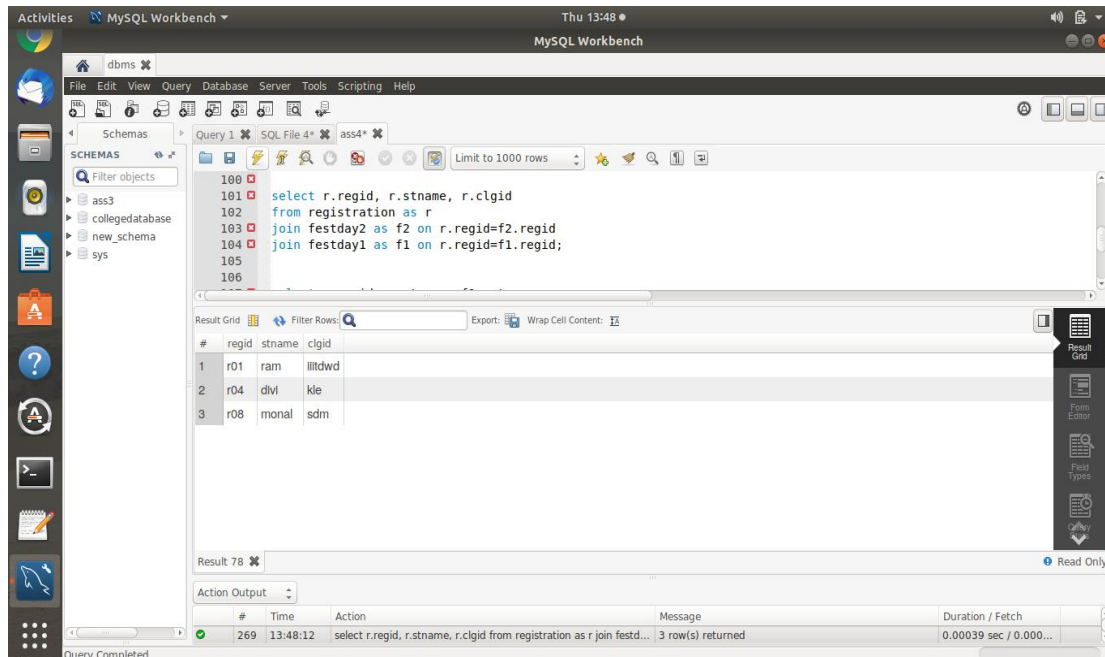
The Action Output pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
267	13:46:39	select f1.regid, f1.stname, f1.entry, f2.entry from festday1 as f1 ...	6 row(s) returned	0.00046 sec / 0.000...

Query Completed

IMPLEMENTING QUERY WITH 2JOINS:

Finding students who have come to fest on both the days and registered using 2joins



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
100 select r.regid, r.stname, r.clgid
101 from registration as r
102 join festday2 as f2 on r.regid=f2.regid
103 join festday1 as f1 on r.regid=f1.regid;
104
105
106
```

The Result Grid displays the following data:

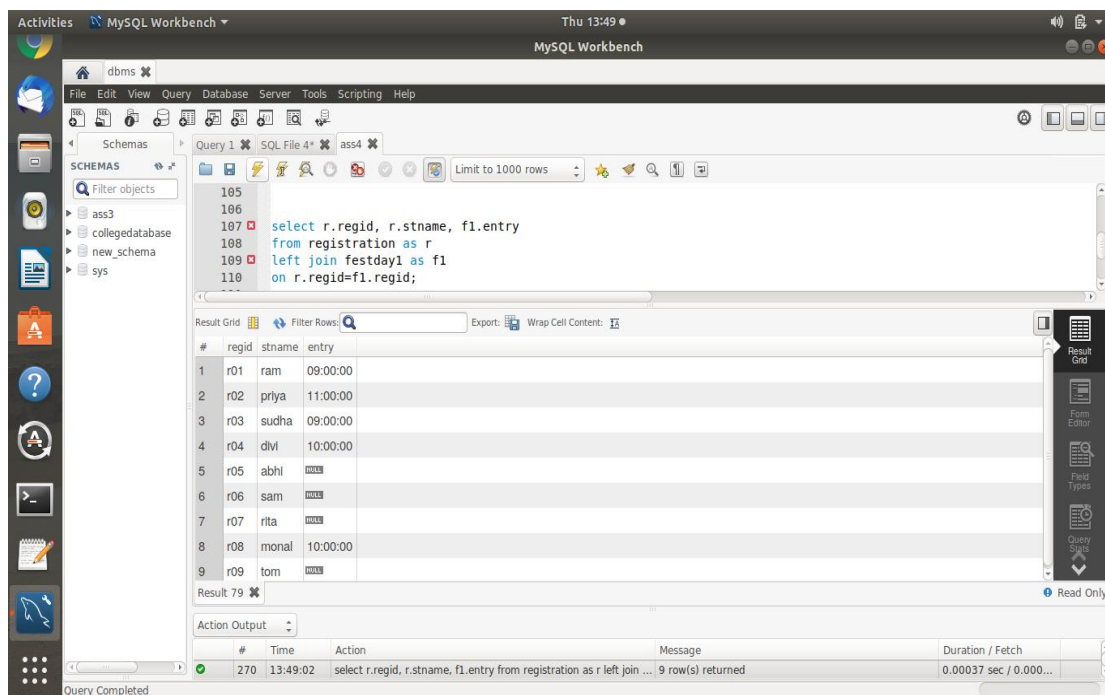
#	regid	stname	clgid
1	r01	ram	lltdwd
2	r04	divi	kle
3	r08	monal	sdm

The Action Output pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
269	13:48:12	select r.regid, r.stname, r.clgid from registration as r join festd...	3 row(s) returned	0.00039 sec / 0.000...

IMPLEMENTING LEFT JOIN:

Finding all students who registered and knowing who came and not came on day1 among them using left join



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
105
106
107 select r.regid, r.stname, f1.entry
108 from registration as r
109 left join festday1 as f1
110 on r.regid=f1.regid;

```

The Result Grid displays the following data:

#	regid	stname	entry
1	r01	ram	09:00:00
2	r02	priya	11:00:00
3	r03	sudha	09:00:00
4	r04	divi	10:00:00
5	r05	abhi	NULL
6	r06	sam	NULL
7	r07	rita	NULL
8	r08	monal	10:00:00
9	r09	tom	NULL

The Action Output pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
270	13:49:02	select r.regid, r.stname, f1.entry from registration as r left join ...	9 row(s) returned	0.00037 sec / 0.000...

Finding all students who registered and knowing who came and not came on day2 among them using left join

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
116
117 • select r.regid, r.stname, f2.entry
118 from registration as r
119 left join festday1 as f2
120 on r.regid=f2.regid;
121
```

The result grid displays 9 rows of data:

#	regid	stname	entry
1	r01	ram	09:00:00
2	r02	priya	11:00:00
3	r03	sudha	09:00:00
4	r04	divi	10:00:00
5	r05	abhi	NULL
6	r06	sam	NULL
7	r07	rita	NULL
8	r08	monal	10:00:00
9	r09	torn	NULL

The Action Output pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
271	13:49:58	select r.regid, r.stname, f2.entry from registration as r left join ...	9 row(s) returned	0.00032 sec / 0.000...

IMPLEMENTING RIGHT JOIN:

Finding students who are registered before among of all the students attended fest on day1 using right join

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
112 select r.stname, f1.entry
113 from registration as r
114 right join festday1 as f1
115 on r.regid=f1.regid;
116
```

The result grid displays 10 rows of data:

#	stname	entry
1	ram	09:00:00
2	priya	11:00:00
3	sudha	09:00:00
4	divi	10:00:00
5	monal	10:00:00
6	NULL	11:00:00
7	NULL	11:00:00
8	NULL	11:00:00
9	NULL	12:00:00
10	NULL	12:00:00

The Action Output pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
272	13:51:20	select r.stname, f1.entry from registration as r right join festda...	10 row(s) returned	0.00033 sec / 0.000...

Finding students who are registered before among of all the students attended fest on day2 using right join

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

123 select r.stname, f2.entry
124 from registration as r
125 right join festday1 as f2
126 on r.regid=f2.regid;
127

```

The Result Grid displays the following data:

#	stname	entry
1	ram	09:00:00
2	priya	11:00:00
3	sudha	09:00:00
4	divi	10:00:00
5	monal	10:00:00
6	NULL	11:00:00
7	NULL	11:00:00
8	NULL	11:00:00
9	NULL	12:00:00
10	NULL	12:00:00

The Action Output shows the query completed successfully, returning 10 rows.

IMPLEMENTING JOIN USING “ GROUPBY” CLAUSE:

Finding no:of students attended to fest from each college out of all registered students using groupby in join

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

128 select r.clgid, count(f1.regid) as 'totalstu'
129 from registration as r
130 join festday1 as f1
131 on r.regid=f1.regid
132 group by r.clgid;
133

```

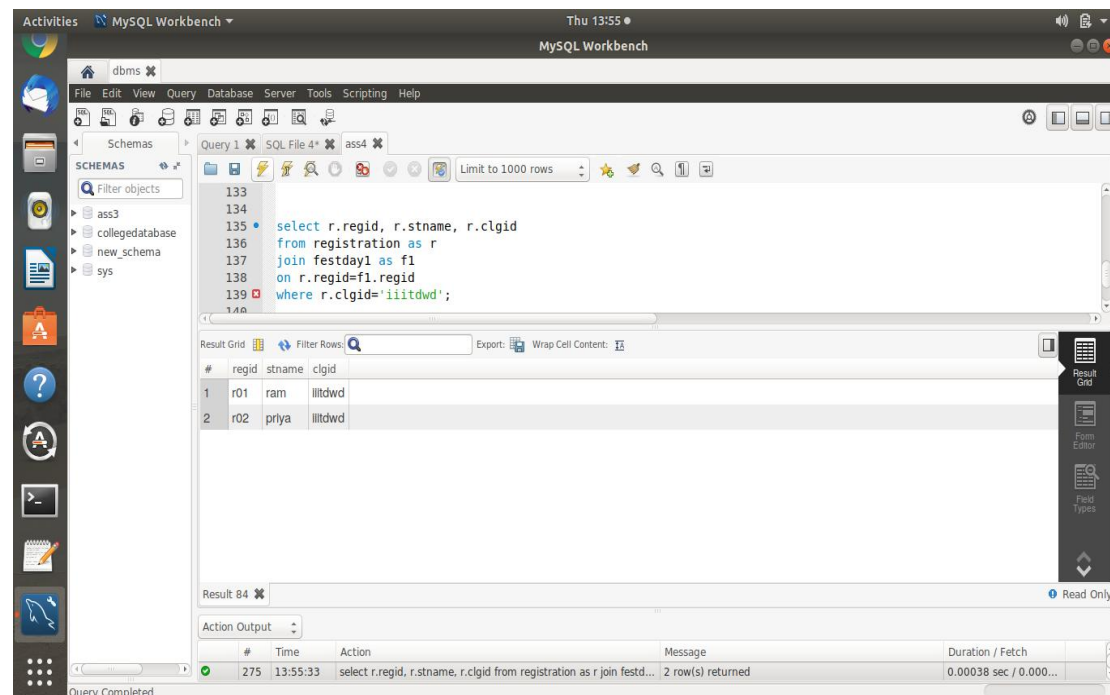
The Result Grid displays the following data:

#	clgid	totalstu
1	liltwdw	2
2	kle	1
3	sdm	2

The Action Output shows the query completed successfully, returning 3 rows.

IMPLEMENTING JOIN USING WHERE CLAUSE:

Finding students who have registered and came to fest on day1 from IIITDWD college using where



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
133
134
135 • select r.regid, r.stname, r.clgid
136 from registration as r
137 join festday1 as f1
138 on r.regid=f1.regid
139 where r.clgid='iiitdwd';
```

The Result Grid displays the following data:

#	regid	stname	clgid
1	r01	ram	iiitdwd
2	r02	priya	iiitdwd

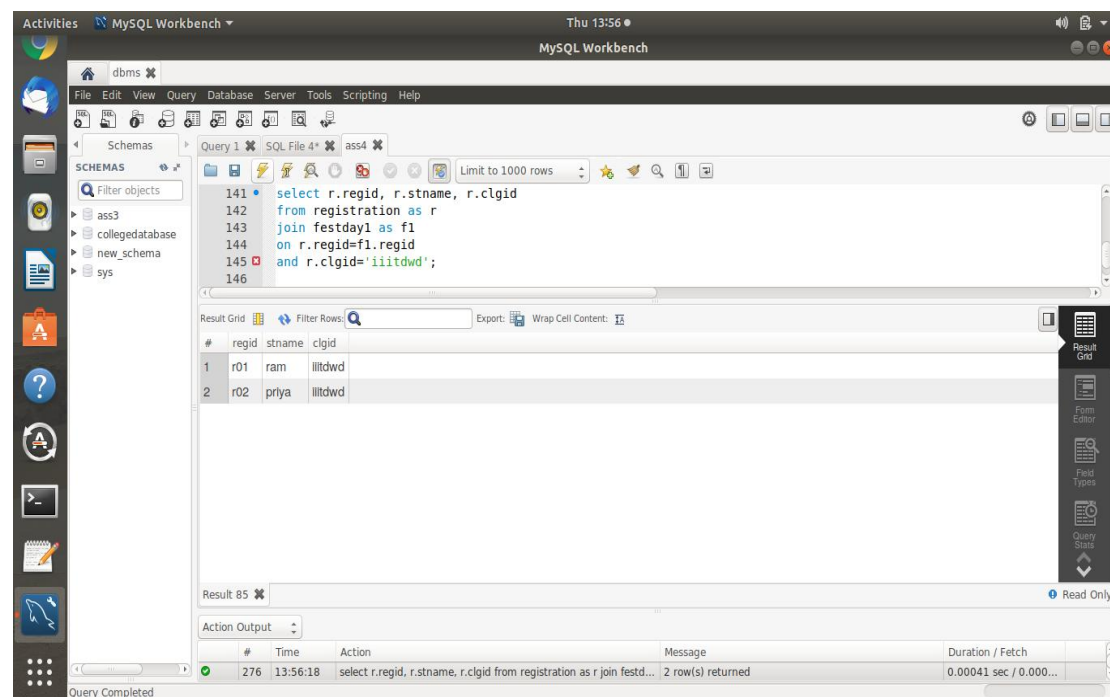
The Action Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
275	13:55:33	select r.regid, r.stname, r.clgid from registration as r join festd...	2 row(s) returned	0.00038 sec / 0.000...

Query Completed

IMPLEMENTING JOIN USING AND:

Finding students who have registered and came to fest on day1 from IIITDWD college using AND operator in join



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
141 • select r.regid, r.stname, r.clgid
142 from registration as r
143 join festday1 as f1
144 on r.regid=f1.regid
145 and r.clgid='iiitdwd';
146
```

The Result Grid displays the following data:

#	regid	stname	clgid
1	r01	ram	iiitdwd
2	r02	priya	iiitdwd

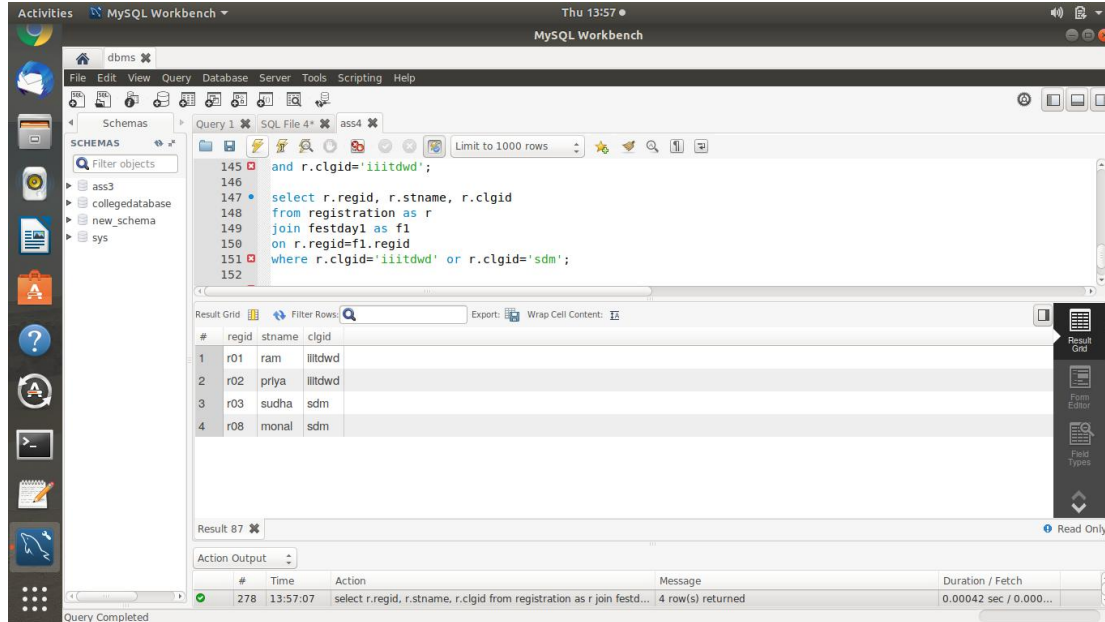
The Action Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
276	13:56:18	select r.regid, r.stname, r.clgid from registration as r join festd...	2 row(s) returned	0.00041 sec / 0.000...

Query Completed

IMPLEMENTING JOIN USING OR:

Finding students who have registered and came to fest on day1 from IIITDWD or SDM college using OR operator in join query



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
145 and r.clgid='iiitdwd';
146
147 select r.regid, r.stname, r.clgid
148 from registration as r
149 join festday1 as f1
150 on r.regid=f1.regid
151 where r.clgid='iiitdwd' or r.clgid='sdm';
152
```

The Result Grid displays the following data:

#	regid	stname	clgid
1	r01	ram	iiitdwd
2	r02	priya	iiitdwd
3	r03	sudha	sdm
4	r08	monal	sdm

The Action Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
278	13:57:07	select r.regid, r.stname, r.clgid from registration as r join festd...	4 row(s) returned	0.00042 sec / 0.000...

---THE_END---