

LAB ASSIGNMENT
DATABASE LAB
TKM COLLEGE OF ENGINEERING

NAME: JOYAL JOSEPH

ROLL NO:224

MANDATORY ASSIGNMENT

```
CREATE database db_aggemp;

use db_aggemp;

CREATE table tbl_details
(e_id int(10)primary key,
e_name varchar(15) not null,
age int(3),
salary float(15));

insert into tbl_details values(101,"Anu",22,9000);
insert into tbl_details values(102,"Shane",29,8000);
insert into tbl_details values(103,"Roshan",34,6000);
insert into tbl_details values(104,"Scott",44,10000);
insert into tbl_details values(105,"Tiger",35,8000);
insert into tbl_details values(106,"Alex",27,7000);
insert into tbl_details values(107,"Abhi",29,8000);

1.select * from tbl_details;
```

Query 1 x

Limit to 1000 rows

```

10 • select * from tbl_details;
11
12
13
14

```

Result Grid

	e_id	e_name	age	salary
▶	101	Anu	22	9000
	102	Shane	29	8000
	103	Roshan	34	6000
	104	Scott	44	10000
	105	Tiger	35	8000
	106	Alex	27	7000

tbl_details 8 x

Output

2. select count(e_name) from tbl_details;

Query 1 x

Limit to 1000 rows

```

10 • insert into tbl_details values(107,"Abhi",29,8000);
11 • select * from tbl_details;
12 • select count(e_name ) from tbl_details;

```

Result Grid

	count(e_name)
▶	7

Result 3 x

Output

Action Output

3. select max(age) from tbl_details;

Query 1 x

Limit to 1000 rows

```

11 • select * from tbl_details;
12 • select count(e_name ) from tbl_details;
13 • select e_name,max(age) from tbl_details;

```

Result Grid

	e_name	max(age)
▶	Anu	44

Result 4 x

Output

Action Output

4. select min(age) from tbl_details;

The screenshot shows the SQL Developer interface. The query editor contains two lines of SQL code:

```
1 • use db_aggemp;  
2 • select min(age) from tbl_details;
```

The toolbar includes icons for file operations, query execution, and a dropdown menu set to "Limit to 1000 rows". Below the query editor, the "Result Grid" tab is active, displaying the following result:

min(age)
44

Below the result grid, the "Output" tab is visible, showing "Result 41" and "Action Output".

5. select sum(age) from tbl_details;

The screenshot shows the SQL Developer interface. The query editor contains two lines of SQL code:

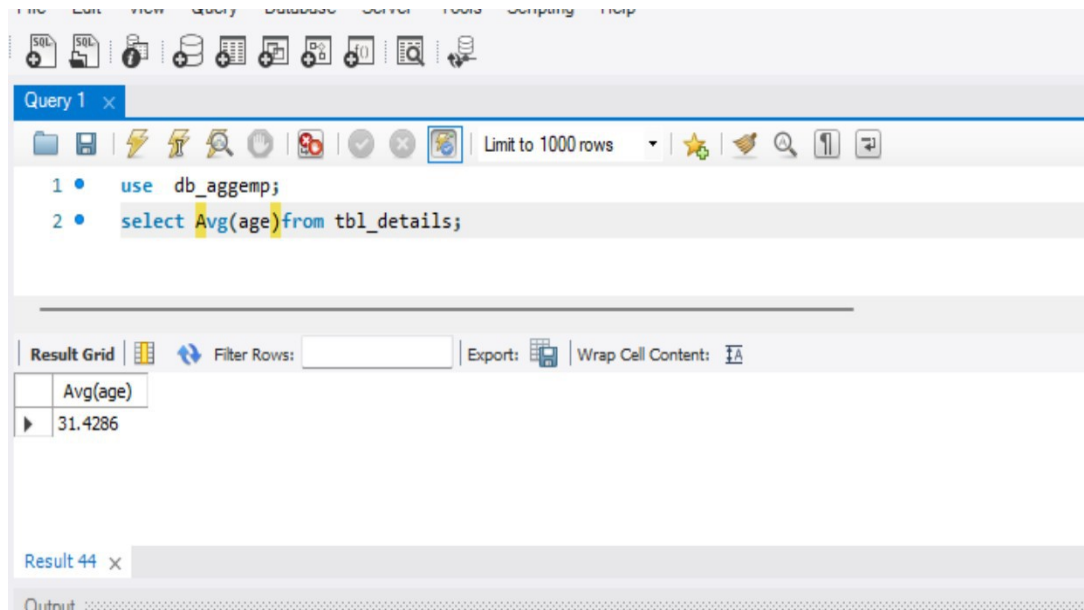
```
1 • use db_aggemp;  
2 • select sum(age) from tbl_details;
```

The toolbar includes icons for file operations, query execution, and a dropdown menu set to "Limit to 1000 rows". Below the query editor, the "Result Grid" tab is active, displaying the following result:

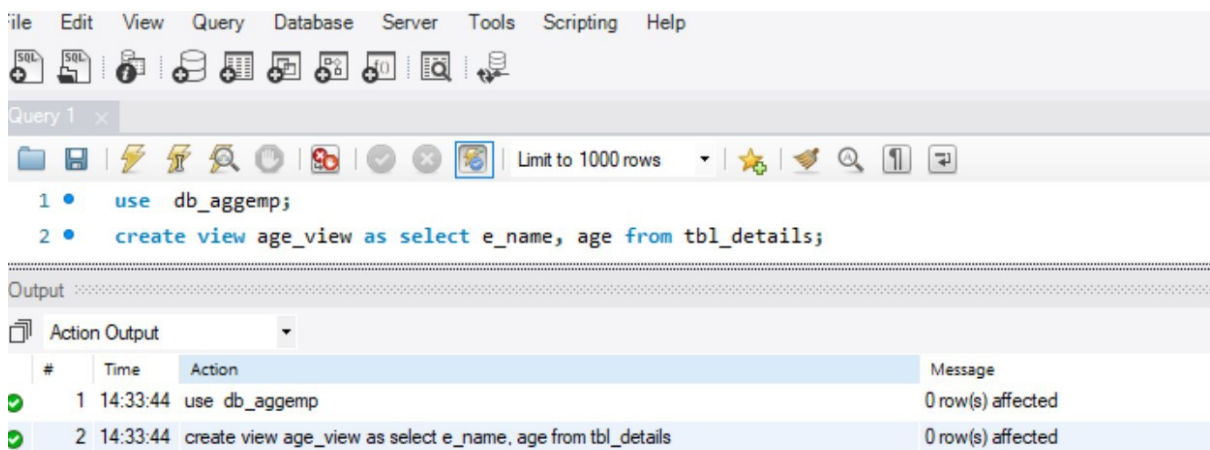
sum(age)
220

Below the result grid, the "Output" tab is visible, showing "Result 43" and "Action Output".

6. select Avg(age) from tbl_details;



7. create view age_view as select e_name, age from tbl_details;



8. select *from details_view;

File Edit View Query Database Server Tools Scripting Help

Query 1 x

Limit to 1000 rows

```
1 • use db_aggemp;
2 • select * from age_view;
```

Result Grid

	e_name	age
▶	Anu	22
	Shane	29
	Roshan	34
	Scott	44
	Tiger	35

age_view 1 x

Output

Action Output

#	Time	Action	Message
✓ 1	14:33:44	use db_aggemp	0 row(s) affected

9. select e_name , salary from tbl_details group by e_name;

File Edit View Query Database Server Tools Scripting Help

Query 1 x

Limit to 1000 rows

```

1 • use db_aggemp;
2 • select e_name , salary from tbl_details group by e_name;
3

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	e_name	salary
▶	Abhi	8000
	Alex	7000
	Anu	9000
	Roshan	6000
	Scott	10000

tbl_details 1 x

Output

Action Output

#	Time	Action	Message
✖ 1	17:11:20	select e_name , salary from tbl_details group by e_name LIMIT 0, 1000	Error Co

10. select e_name , salary from tbl_details order by salary asc; OR

select e_name , salary from tbl_details order by salary ;

Query 1 x

Limit to 1000 rows

```

1 • use db_aggemp;
2 • select * from age_view;insert into tbl_details values(101,"Anu",22,9000);
3 • create view age_view as select e_name, age from tbl_details;
4 • select e_name, salary from tbl_details group by e_name;
5 • select e_name , salary from tbl_details order by salary asc;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	e_name	salary
	Roshan	6000
	Alex	7000
	Shane	8000
	Tiger	8000
	Abhi	8000

tbl_details 6 x

Output

Action Output

11. select e_name , salary from tbl_details order by salary DESC;

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The script area contains the following SQL queries:

```

2 • select * from age_view; insert into tbl_details values(101, "Anu", 22, 9000);
3 • create view age_view as select e_name, age from tbl_details;
4 • select e_name, salary from tbl_details group by e_name;
5 • select e_name, salary from tbl_details order by salary asc;
6 • select e_name, salary from tbl_details order by salary desc;
7

```

Below the script is the 'Result Grid' section, which includes a 'Filter Rows' input, an 'Export' button, and a 'Wrap Cell Content' checkbox. The grid displays the following data:

e_name	salary
Scott	10000
Anu	9000
Shane	8000
Tiger	8000
Abhi	8000
Alex	7000

At the bottom, there is a tab labeled 'tbl_details 7' and an 'Output' section.

GROUP 4

AIM:

- (i) Consider the database for a college and design an ER diagram. Write the query for the following.
- (ii) Create the tables:
- (iii)
- (iv) Student (sid, sname, sex, dob, dno) Department (dno, dname)
- (v) Faculty (F_id, fname, designation, salary, dno) Course (cid, cname, credits, dno)
- (vi) Register(sid, cid, sem) Teaching(f_id, cid, sem) Hostel(hid, hname, seats,)
- (vii) Include the necessary constraints NOT NULL, DEFAULT, CHECK, and PRIMARY KEY, UNIQUE.
- (viii)
 - (1.i) Create a database college
- (ix)
 - (1.i) Use college as the current database

- (1.ii) Display all the tables in collegedatabase
- (x) (1.i) Describe the structure of all tables
- (xi) (1.i) Modify the student table to add a new field 'grade'
- (xii) Consider the database for a college. Write the query for the following.
- (xiii) 1. Insert at least 5 tuples into each table.
- (xiv) 1. List the details of students in the ascending order of date of birth
- (xv) 1. Display the details of students from computer department
- (xvi) 1. List the faculties in the descending order of salary
- (xvii) 1. Display the total number of students in each department
- (xviii) 1. Display the total number of faculties in each department with salary greater than 25000

(xix)

(xx) **CODE:**

(xxi)

(xxii) create database college;

(xxiii) use college;

(xxiv)

(xxv) create table department(dno varchar(20) not null,dname varchar(20) not null,primary key(dno));

(xxvi) create table student(sid varchar(20) not null,sname varchar(20) not null,sex varchar(10) not null,dob date not null,dno varchar(20) not null,primary key(sid),foreign key(dno) references department(dno) on delete cascade);

(xxvii) create table faculty(fid varchar(20) not null,fname varchar(30) not null,designation varchar(20) not null,salary integer(10) check (salary>5000 and salary <=150000),dno varchar(20) not null,primary key(fid),foreign key(dno) references department(dno) on delete cascade);

(xxviii) create table course(cid varchar(20) not null,cname varchar(20) not null,credits integer(10) not null,dno varchar(20) not null,primary key(cid),foreign key(dno) references department(dno) on delete cascade);

(xxix) create table register(sid varchar(20) not null,cid varchar(20) not null,sem varchar(10) not null,foreign key(sid) references student(sid) on delete cascade,foreign key(cid) references course(cid) on delete cascade);

(xxx) create table teaching(fid varchar(20) not null,cid varchar(20) not null,sem varchar(10) not null,foreign key(fid) references faculty(fid) on delete cascade,foreign key(cid) references course(cid) on delete cascade);

(xxxi) create table hostel(hid varchar(20) not null,hname varchar(20) not null default "modern hostel",seats integer(20) not null,unique(hid));

(xxxii)

(xxxiii) insert into department values("d001","computer");

(xxxiv) insert into department values("d002","maths");

(xxxv) insert into department values("d003","english");

(xxxvi) insert into department values("d004","science");

(xxxvii) insert into department values("d005","humanities");

(xxxviii) insert into student values("s001","sharanya","f","1998-12-26","d001");

(xxxix) insert into student values("s002","sham","m","1997-01-16","d002");

(xl) insert into student values("s003","jacob","m","1998-12-06","d001");

(xli) insert into student values("s004","cathy","f","1998-11-23","d003");

(xlii) insert into student values("s005","ajay","m","1998-03-03","d004");

(xliii) insert into faculty values("f001","manav","associate professor",50000,"d001");

(xliv) insert into faculty values("f002","asha","associate professor",40000,"d002");

(xlv) insert into faculty values("f003","sree","asisstant professor",100000,"d003");

(xlvi) insert into faculty values("f004","akash","associate professor",50000,"d004");

(xlvii) insert into faculty values("f005","lena","associate professor",50000,"d005");

(xlviii) insert into course values("c001","differentiation",10,"d002");

(xlix) insert into course values("c002","java",10,"d001");

(l) insert into course values("c003","anatomy",10,"d004");

(li) insert into course values("c004","grammar",5,"d003");

(lii) insert into course values("c005","networks",10,"d001");

(liii) insert into course values("c006","basics humanities",10,"d005");

(liv) insert into register values("s001","c002","1");

(lv) insert into register values("s001","c005","1");

(lvi) insert into register values("s002","c001","2");

(lvii) insert into register values("s005","c003","1");

(lviii) insert into register values("s004","c004","2");

(lix) insert into register values("s003","c002","1");

(lx) insert into register values("s006","c006","1");

(lxi) insert into teaching values("f001","c002","1");

(lxii) insert into teaching values("f002","c001","2");

(lxiii) insert into teaching values("f003","c004","2");

(lxiv) insert into teaching values("f004","c003","1");

(lxv) insert into teaching values("f005","c006","1");

(lxvi) insert into hostel values("h001","rose hostel",20);

(lxvii) insert into hostel values("h002","general hostel",20);

(lxviii) insert into hostel values("h003","boys hostel",20);

(lxix) insert into hostel values("h004","working hostel",20);

(lxx) insert into hostel values("h005","mixed hostel",25);

(lxxi)

(lxxii) select * from department;

(lxxiii) select * from student;

(lxxiv) select * from faculty;

(lxxv) select * from course;

(lxxvi) select * from teaching;

(lxxvii) select * from register;

(lxxviii)

(lxxix) alter table student add grade varchar(5) not null;

(lxxx) insert into student values("s006","lalitha","f","1998-04-08","d005","A");

(lxxxi) update student set grade="A" where sid="s001";

(lxxxii) update student set grade="A" where sid="s002";

(lxxxiii) update student set grade="B" where sid="s003";

(lxxxiv) update student set grade="B+" where sid="s004";

(lxxxv) update student set grade="A+" where sid="s005";

(lxxxvi)

(lxxxvii) select * from student order by dob;

(lxxxviii)

(lxxxix) select * from student where dno="d001";

(xc)

(xci) select * from faculty order by salary desc;

(xcii)

(xciii) select count(sid) as "no of student",dno from student group by dno;

(xciv)

(xcv) select count(fid) as "no of faculty",dno from faculty group by dno having salary>=25000;

(xcvi)

(xcvii)

(xcviii)

(xcix) **OUTPUT:**

(c)

(ci)

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

```
49 select * from student;
50 select * from faculty;
51 select * from course;
52 select * from teaching;
53 select * from register;
54 alter table student add grade varchar(5) not null;
55 insert into student values("c006","lalitha","f","1998-04-08","s005","A");
```

The result grid displays the following data:

sid	sname	sex	dob	dno	grade
s001	sharanya	f	1998-12-26	d001	A
s002	sham	m	1997-01-16	d002	A
s003	jacob	m	1998-12-06	d001	B
s004	cathy	f	1998-11-23	d003	B+
s005	ajay	m	1998-03-03	d004	A+

The bottom panel shows the Action Output with the following log:

#	Time	Action	Message	Duration / Fetch
75	08:41:11	insert into student values("s005","lalitha","f","1998-04-08","s005","A")	1 row(s) affected	0.016 sec
76	08:41:37	insert into course values("c006","basics humanities",10,"d005")	1 row(s) affected	0.016 sec
77	08:41:46	insert into register values("s006","c006","1")	1 row(s) affected	0.000 sec
78	08:41:56	insert into teaching values("t005","c006","1")	1 row(s) affected	0.016 sec
79	08:42:02	insert into hostel values("h005","mixed hostel",25)	1 row(s) affected	0.000 sec
80	08:42:31	select * from student LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

(cii)

(ciii)

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab is active, displaying a SQL script. The 'Result Grid' shows the output of the query, which is a table with 5 columns: fid, fname, designation, salary, and dno. The table contains 5 rows of data. The 'Output' pane shows the execution log, indicating that the query was successful and returned 5 rows.

```
49 select * from student;
50 select * from faculty;
51 select * from course;
52 select * from teaching;
53 select * from register;
54 alter table student add grade varchar(5) not null;
55 insert into student values('c006','Jalitha','F','1000-AJ-00','AJAC','A');
```

fid	fname	designation	salary	dno
f001	manav	associate professor	50000	d001
f002	asha	associate professor	40000	d002
f003	sree	assistant professor	100000	d003
f004	alash	associate professor	50000	d004
f005	lena	associate professor	50000	d005

Output

#	Time	Action	Message	Duration / Fetch
76	08:41:37	insert into course values('c005','basics humanities','10','d005')	1 row(s) affected	0.016 sec
77	08:41:46	insert into register values('c006','c006','1')	1 row(s) affected	0.000 sec
78	08:41:56	insert into teaching values('1005','c006','1')	1 row(s) affected	0.016 sec
79	08:42:02	insert into hostel values('h005','mixed hostel',25)	1 row(s) affected	0.000 sec
80	08:42:31	select * from student LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
81	08:43:11	select * from faculty LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

(civ)

(cv)

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab is active, displaying a SQL script. The 'Result Grid' shows the output of the query, which is a table with 5 columns: cid, cname, credits, and dno. The table contains 5 rows of data. The 'Output' pane shows the execution log, indicating that the query was successful and returned 5 rows.

```
49 select * from student;
50 select * from faculty;
51 select * from course;
52 select * from teaching;
53 select * from register;
54 alter table student add grade varchar(5) not null;
55 insert into student values('c006','Jalitha','F','1000-AJ-00','AJAC','A');
```

cid	cname	credits	dno
c001	differentiation	10	d002
c002	java	10	d001
c003	anatomy	10	d004
c004	grammar	5	d003
c005	networks	10	d001

Output

#	Time	Action	Message	Duration / Fetch
77	08:41:46	insert into register values('c006','c006','1')	1 row(s) affected	0.000 sec
78	08:41:56	insert into teaching values('1005','c006','1')	1 row(s) affected	0.016 sec
79	08:42:02	insert into hostel values('h005','mixed hostel',25)	1 row(s) affected	0.000 sec
80	08:42:31	select * from student LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
81	08:43:11	select * from faculty LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
82	08:43:21	select * from course LIMIT 0, 1000	6 row(s) returned	0.016 sec / 0.000 sec

(cvi)

(cvii)

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

```
49 select * from student;
50 -- Execute the selected portion of the script or everything, if there is no selection
51 select * from course;
52 select * from teaching;
53 select * from register;
54 alter table student add grade varchar(5) not null;
55 insert into student values('c006','Jalitha','F','1000-AJ-AA','AAAC','A');
```

The result grid shows the following data:

fid	cid	sem
r002	c001	2
r003	c004	2
r004	c003	1
r001	c002	1
r005	c006	1

The output window shows the following action output:

#	Time	Action	Message	Duration / Fetch
78	08:41:56	insert into teaching values('1005','c006','1');	1 row(s) affected	0.016 sec
79	08:42:02	insert into hostel values('h005','mixed hostel',25);	1 row(s) affected	0.000 sec
80	08:42:31	select * from student LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
81	08:43:11	select * from faculty LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
82	08:43:21	select * from course LIMIT 0, 1000	6 row(s) returned	0.016 sec / 0.000 sec
83	08:43:48	select * from teaching LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

(cviii)

(cix)

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

```
49 select * from student;
50 select * from faculty;
51 select * from course;
52 select * from teaching;
53 select * from register;
54 alter table student add grade varchar(5) not null;
55 insert into student values('c006','Jalitha','F','1000-AJ-AA','AAAC','A');
```

The result grid shows the following data:

sid	cid	sem
s001	c002	1
s001	c005	1
s002	c001	2
s005	c003	1
s004	c004	2

The output window shows the following action output:

#	Time	Action	Message	Duration / Fetch
78	08:42:02	insert into hostel values('h005','mixed hostel',25);	1 row(s) affected	0.000 sec
80	08:42:31	select * from student LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
81	08:43:11	select * from faculty LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
82	08:43:21	select * from course LIMIT 0, 1000	6 row(s) returned	0.016 sec / 0.000 sec
83	08:43:48	select * from teaching LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
84	08:43:53	select * from register LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

(cx)

(cxi)

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

```
56 • update student set grade="A" where sid="s001";
57 • update student set grade="A" where sid="s002";
58 • update student set grade="B" where sid="s003";
59 • update student set grade="B+" where sid="s004";
60 • update student set grade="A+" where sid="s005";
61 • select * from student order by dob;
```

The result grid displays the following data:

sid	sname	sex	dob	dno	grade
s002	sham	m	1997-01-16	d002	A
s005	ajay	m	1998-03-03	d004	A+
s006	lalitha	f	1998-04-08	d005	A
s004	cathy	f	1998-11-23	d003	B+
s003	jacob	m	1998-12-06	d001	B

The bottom panel shows the Action Output table with the following data:

#	Time	Action	Message	Duration / Fetch
80	08:42:31	select * from student LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
81	08:43:11	select * from faculty LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
82	08:43:21	select * from course LIMIT 0, 1000	6 row(s) returned	0.016 sec / 0.000 sec
83	08:43:48	select * from teaching LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
84	08:43:53	select * from register LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
85	08:46:41	select * from student order by dob LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

(cxii)

(cxiii)

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

```
57 • update student set grade="A" where sid="s002";
58 • update student set grade="B" where sid="s003";
59 • update student set grade="B+" where sid="s004";
60 • update student set grade="A+" where sid="s005";
61 • select * from student order by dob;
62 • select * from student where dno="d001";
```

The result grid displays the following data:

sid	sname	sex	dob	dno	grade
s001	sharanya	f	1998-12-26	d001	A
s003	jacob	m	1998-12-06	d001	B

The bottom panel shows the Action Output table with the following data:

#	Time	Action	Message	Duration / Fetch
81	08:43:11	select * from faculty LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
82	08:43:21	select * from course LIMIT 0, 1000	6 row(s) returned	0.016 sec / 0.000 sec
83	08:43:48	select * from teaching LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
84	08:43:53	select * from register LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
85	08:46:41	select * from student order by dob LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
86	08:48:08	select * from student where dno="d001" LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

(cxiv)

(cxv)

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

```
58 • update student set grade="B" where sid="s003";
59 • update student set grade="B+" where sid="s004";
60 • update student set grade="A+" where sid="s005";
61 • select * from student order by dob;
62 • select * from student where dno="d001";
63 • select * from faculty order by salary desc;
```

The result grid displays the output of the last query, showing a list of faculty members with columns: fid, fname, designation, salary, and dno.

fid	fname	designation	salary	dno
f003	sree	assistant professor	100000	d003
f001	manav	associate professor	50000	d001
f004	akash	associate professor	50000	d004
f005	lena	associate professor	50000	d005
f002	ashu	associate professor	40000	d002

The bottom panel shows the Action Output window with a table of query execution details:

#	Time	Action	Message	Duration / Fetch
82	08:43:21	select "from course LIMIT 0, 1000	6 row(s) returned	0.016 sec / 0.000 sec
83	08:43:48	select "from teaching LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
84	08:43:53	select "from register LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
85	08:46:41	select "from student order by dob LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
86	08:48:08	select "from student where dno="d001" LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
87	08:48:43	select "from faculty order by salary desc LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

(cxvi)

(cxvii)

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

```
58 • update student set grade="B" where sid="s003";
59 • update student set grade="B+" where sid="s004";
60 • update student set grade="A+" where sid="s005";
61 • select * from student order by dob;
62 • select * from student where dno="d001";
63 • select * from faculty order by salary desc;
64 • select count(sid) as "no of student",dno from student group by dno;
```

The result grid displays the output of the last query, showing a list of student counts grouped by department (dno).

no of student	dno
2	d001
1	d002
1	d003
1	d004
1	d005

The bottom panel shows the Action Output window with a table of query execution details:

#	Time	Action	Message	Duration / Fetch
85	08:46:41	select "from student order by dob LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
86	08:48:08	select "from student where dno="d001" LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
87	08:48:43	select "from faculty order by salary desc LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec