

Part 1 .Write following simple SQL Queries on the University Schema.

A.Find the names of all the students whose total credits are greater than 100.

```
select name from student where tot_cred > 100;
```

The screenshot shows the MySQL Workbench interface. The main editor displays a series of SQL queries for inserting data into the 'time_slot' table. Below the editor, the 'Result Grid' shows the results of the last query, which is a SELECT statement. The results are displayed in a table with one column, 'name', and three rows: Zhang, Chavez, and Tanaka. The bottom panel shows the 'Output' tab with a table of execution logs.

SQL Queries:

```
228 • insert into time_slot values ('D', 'W', '13', '0', '13', '50');
229 • insert into time_slot values ('D', 'F', '13', '0', '13', '50');
230 • insert into time_slot values ('E', 'T', '10', '30', '11', '45 ');
231 • insert into time_slot values ('E', 'R', '10', '30', '11', '45 ');
232 • insert into time_slot values ('F', 'T', '14', '30', '15', '45 ');
233 • insert into time_slot values ('F', 'R', '14', '30', '15', '45 ');
234 • insert into time_slot values ('G', 'M', '16', '0', '16', '50');
235 • insert into time_slot values ('G', 'W', '16', '0', '16', '50');
```

Result Grid:

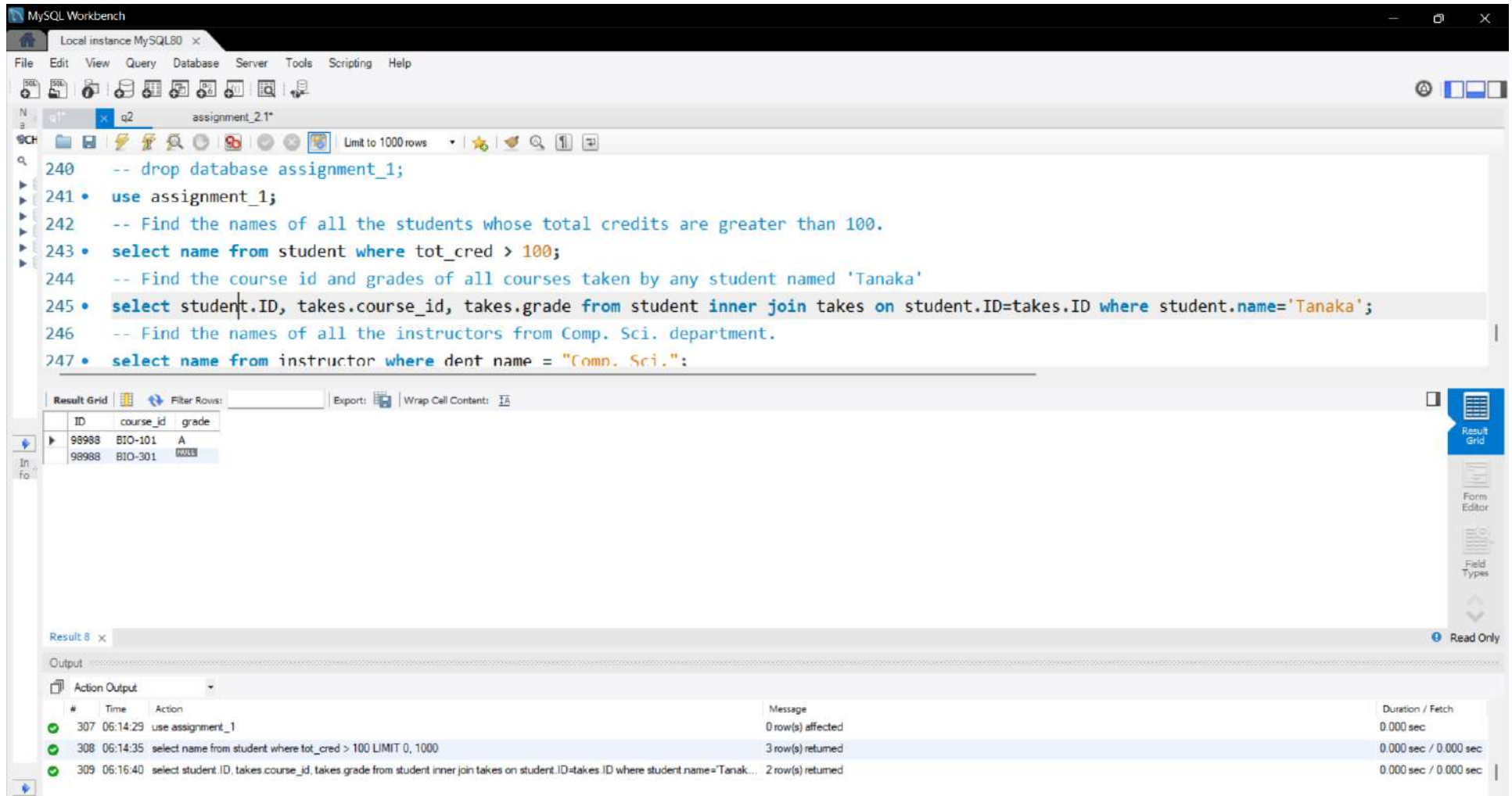
name
Zhang
Chavez
Tanaka

Output:

#	Time	Action	Message	Duration / Fetch
306	06:14:17	select course.course_id, course.title from course inner join section on section.course_id = course.course_id where semester in (F...	13 row(s) returned	0.000 sec / 0.000 sec
307	06:14:29	use assignment_1	0 row(s) affected	0.000 sec
308	06:14:35	select name from student where tot_cred > 100 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

B.Find the course id and grades of all courses taken by any student named 'Tanaka'

select student.ID, takes.course_id, takes.grade from student inner join takes on student.ID=takes.ID where student.name='Tanaka';



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

```

240 -- drop database assignment_1;
241 • use assignment_1;
242 -- Find the names of all the students whose total credits are greater than 100.
243 • select name from student where tot_cred > 100;
244 -- Find the course id and grades of all courses taken by any student named 'Tanaka'
245 • select student.ID, takes.course_id, takes.grade from student inner join takes on student.ID=takes.ID where student.name='Tanaka';
246 -- Find the names of all the instructors from Comp. Sci. department.
247 • select name from instructor where dept_name = "Comp. Sci.";
  
```

The results of the query in line 245 are displayed in the Result Grid:

ID	course_id	grade
98988	BIO-101	A
98988	BIO-301	NULL

The Output tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
307	06:14:29	use assignment_1	0 row(s) affected	0.000 sec
308	06:14:35	select name from student where tot_cred > 100 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
309	06:16:40	select student.ID, takes.course_id, takes.grade from student inner join takes on student.ID=takes.ID where student.name='Tanaka'	2 row(s) returned	0.000 sec / 0.000 sec

C. Find the names of all the instructors from Comp. Sci. department.

select name from instructor where dept_name = "Comp. Sci.";

The screenshot shows the MySQL Workbench interface. The main editor displays a SQL script with several queries. The query at line 247 is highlighted: `select name from instructor where dept_name = "Comp. Sci.";`. Below the editor, the 'Result Grid' shows the results of this query, displaying three rows: Srinivasan, Katz, and Brandt. The 'Output' pane at the bottom shows the execution log for three queries, including the one that was executed, with details on the number of rows returned and the duration.

SQL Script:

```

242 -- Find the names of all the students whose total credits are greater than 100.
243 • select name from student where tot_cred > 100;
244 -- Find the course id and grades of all courses taken by any student named 'Tanaka'
245 • select student.ID, takes.course_id, takes.grade from student inner join takes on student.ID=takes.ID where student.name='Tanaka';
246 -- Find the names of all the instructors from Comp. Sci. department.
247 • select name from instructor where dept_name = "Comp. Sci.";
248 -- Find the course id and titles of all courses taught by an instructor named 'Srinivasan'
249 • select course.course_id, course.title from course inner join teaches on teaches.course_id = course.course_id where teaches.ID in (select inst

```

Result Grid:

name
Srinivasan
Katz
Brandt

Output Log:

#	Time	Action	Message	Duration / Fetch
308	06:14:35	select name from student where tot_cred > 100 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
309	06:16:40	select student.ID, takes.course_id, takes.grade from student inner join takes on student.ID=takes.ID where student.name='Tanaka'	2 row(s) returned	0.000 sec / 0.000 sec
310	06:18:56	select name from instructor where dept_name = "Comp. Sci." LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

D.Find the course id and titles of all courses taught by an instructor named 'Srinivasan'

```
select course.course_id, course.title from course inner join teaches on teaches.course_id = course.course_id where teaches.ID in (select instructor.ID
from instructor where instructor.name='Srinivasan');
```

The screenshot shows the MySQL Workbench interface. The main window displays a SQL script with several queries. The query at line 249 is highlighted, which is the query to find the course id and titles of all courses taught by an instructor named 'Srinivasan'.

```
243 • select name from student where tot_cred > 100;
244 -- Find the course id and grades of all courses taken by any student named 'Tanaka'
245 • select student.ID, takes.course_id, takes.grade from student inner join takes on student.ID=takes.ID where student.name='Tanaka';
246 -- Find the names of all the instructors from Comp. Sci. department.
247 • select name from instructor where dept_name = "Comp. Sci.";
248 -- Find the course id and titles of all courses taught by an instructor named 'Srinivasan'
249 • select course.course_id, course.title from course inner join teaches on teaches.course_id = course.course_id where teaches.ID in (select inst
250 -- Find the names of instructors who have taught at least one course in Spring 2009.
```

Below the script, the 'Result Grid' shows the results of the highlighted query (line 249). The results are as follows:

course_id	title
CS-101	Intro. to Computer Science
CS-315	Robotics
CS-347	Database System Concepts

The 'Output' window at the bottom shows the execution log for the queries:

#	Time	Action	Message	Duration / Fetch
309	06:16:40	select student.ID, takes.course_id, takes.grade from student inner join takes on student.ID=takes.ID where student.name='Tanaka'...	2 row(s) returned	0.000 sec / 0.000 sec
310	06:18:56	select name from instructor where dept_name = "Comp. Sci." LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
311	06:20:14	select course.course_id, course.title from course inner join teaches on teaches.course_id = course.course_id where teaches.ID in ...	3 row(s) returned	0.000 sec / 0.000 sec

E. Find the names of instructors who have taught at least one course in Spring 2009.

`select instructor.name from instructor inner join teaches on teaches.ID = instructor.ID where teaches.semester = 'Spring' and year = 2009;`

The screenshot shows the MySQL Workbench interface. The SQL editor contains several queries, with the following query selected and highlighted:

```
251 • select instructor.name from instructor inner join teaches on teaches.ID = instructor.ID where teaches.semester = 'Spring' and year = 2009;
```

The Results panel shows the output of the selected query (Result 11). The output is a table with one column, 'name', and three rows of data:

name
Brandt
Brandt
Kim

The Action Output panel at the bottom shows the execution log for the queries:

#	Time	Action	Message	Duration / Fetch
310	06:18:56	select name from instructor where dept_name = "Comp. Sci." LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
311	06:20:14	select course.course_id, course.title from course inner join teaches on teaches.course_id = course.course_id where teaches.ID in ...	3 row(s) returned	0.000 sec / 0.000 sec
312	06:20:59	select instructor.name from instructor inner join teaches on teaches.ID = instructor.ID where teaches.semester = 'Spring' and year ...	3 row(s) returned	0.000 sec / 0.000 sec

F. Find the courses which are offered in both 'Fall' and 'Spring' semester (not necessarily in the same year).

`select course.course_id, course.title from course inner join section on section.course_id = course.course_id where semester in ('Fall','Spring');`

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

```

247 • select name from instructor where dept_name = "Comp. Sci.";
248 -- Find the course id and titles of all courses taught by an instructor named 'Srinivasan'
249 • select course.course_id, course.title from course inner join teaches on teaches.course_id = course.course_id where teaches.ID in (select inst
250 -- Find the names of instructors who have taught at least one course in Spring 2009.

```

The Results grid shows the following data:

course_id	title
CS-101	Intro. to Computer Science
CS-101	Intro. to Computer Science
FIN-201	Investment Banking
MUJ-199	Music Video Production
HIS-351	World History
CS-190	Game Design
CS-190	Game Design
CS-319	Image Processing
CS-347	Database System Concepts
EE-181	Intro. to Digital Systems
CS-319	Image Processing
PHY-101	Physical Principles
CS-315	Robotics

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
311	06:20:14	select course.course_id, course.title from course inner join teaches on teaches.course_id = course.course_id where teaches.ID in...	3 row(s) returned	0.000 sec / 0.000 sec
312	06:20:59	select instructor.name from instructor inner join teaches on teaches.ID = instructor.ID where teaches.semester = 'Spring' and year ...	3 row(s) returned	0.000 sec / 0.000 sec
313	06:21:34	select course.course_id, course.title from course inner join section on section.course_id = course.course_id where semester in ('Fall','Spring');	13 row(s) returned	0.000 sec / 0.000 sec

Create the Railway schema using the commands in the Railway DDL script; The DDL script also contains a description of the railway Schema.

Insert sample data using the command in the file SampleRailwayData.

Write the following Queries for Railway Schema.

1.Find pairs of stations (station codes) that have a track (direct connection) with distance less than 20Kms between them.

`select stcode1, stcode2 from track where distance < 20;`

The screenshot shows the MySQL Workbench interface. The main editor displays a SQL script with the following content:

```

81
82 -- Create the Railway schema using the commands in the Railway DDL script; The DDL script also contains a description of the railway Schema.
83 -- Insert sample data using the command in the file SampleRailwayData.
84 • use assignment_12;
85 -- Write the following Queries for Railway Schema.
86 -- 1.Find pairs of stations (station codes) that have a track (direct connection) with distance less than 20Kms between them.
87 • select stcode1, stcode2 from track where distance < 20;
88 -- 2.Find the IDs of all the trains which have a stop at THANE
89 • select trainhalts.id from trainhalts inner join station on station.stcode = trainhalts.stcode where station.name = 'THANE';
  
```

The 'Result Grid' shows the output of the query executed in line 87:

stcode1	stcode2
BYC	DR
BYC	KRL
CST	BYC
CST	DR
CST	KRL
GRP	TNA
INDR	INDR

The 'Action Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
380	06:27:54	create database assignment_12	Error Code: 1007. Can't create database 'assignment_12'; database exists	0.000 sec
381	06:28:11	use assignment_12	0 row(s) affected	0.000 sec
382	06:28:15	select stcode1, stcode2 from track where distance < 20 LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

2. Find the IDs of all the trains which have a stop at THANE

`select trainhalts.id from trainhalts inner join station on station.stcode = trainhalts.stcode where station.name = 'THANE';`

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

```

84 • use assignment_12;
85 -- Write the following Queries for Railway Schema.
86 -- 1. Find pairs of stations (station codes) that have a track (direct connection) with distance less than 20Kms between them.
87 • select stcode1, stcode2 from track where distance < 20;
88 -- 2. Find the IDs of all the trains which have a stop at THANE
89 • select trainhalts.id from trainhalts inner join station on station.stcode = trainhalts.stcode where station.name = 'THANE';
90 -- 3. Find the names of all trains that start at MUMBAI.
91 • select train.name from train where train.id in (select trainhalts.id from trainhalts inner join station on trainhalts.stcode = station.stcode where station.
92 -- 4. List all the stations in order of visit by the train 'CST-AMR LOCAL'

```

The query results are displayed in the Result Grid, showing the IDs of the trains that stop at THANE:

id
A65
KP11

The Action Output pane shows the execution details of the queries:

#	Time	Action	Message	Duration / Fetch
381	06:28:11	use assignment_12	0 row(s) affected	0.000 sec
382	06:28:15	select stcode1, stcode2 from track where distance < 20 LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
383	06:29:51	select trainhalts.id from trainhalts inner join station on station.stcode = trainhalts.stcode where station.name = 'THANE' LIMIT 0, 10...	2 row(s) returned	0.000 sec / 0.000 sec

3. Find the names of all trains that start at MUMBAI.

select train.name from train where train.id in (select trainhalts.id from trainhalts inner join station on trainhalts.stcode = station.stcode where station.name = "MUMBAI");

The screenshot shows the MySQL Workbench interface. The main editor window displays a SQL query to find train names starting from Mumbai. The query is as follows:

```

91 • select train.name from train where train.id in (select trainhalts.id from trainhalts inner join station on trainhalts.stcode = station.stcode where station.name = "MUMBAI");
92 -- 4. List all the stations in order of visit by the train 'CST-AMR_LOCAL'.
93 • select trainhalts.stcode from trainhalts inner join train on train.id = trainhalts.id where train.name = 'CST-AMR_LOCAL';
94
95 -- drop table trainhalts;
96 -- drop table train;
97 -- drop table track;
98 -- drop table station;
99
100 -- delete from trainhalts;
  
```

Below the query editor, the 'Result Grid' tab is active, showing the results of the first query. The results are as follows:

name
CST-AMR_LOCAL
CST-KYN

The 'Output' tab at the bottom shows the execution log with three entries:

#	Time	Action	Message	Duration / Fetch
65	06:11:46	select train.name from train where train.id in (select trainhalts.id from trainhalts inner join station on trainhalts.stcode = station.stcode where station.name = "MUMBAI");	2 row(s) returned	0.000 sec / 0.000 sec
66	06:11:46	select trainhalts.stcode from trainhalts inner join train on train.id = trainhalts.id where train.name = 'CST-AMR_LOCAL' LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
67	06:11:52	select train.name from train where train.id in (select trainhalts.id from trainhalts inner join station on trainhalts.stcode = station.stcode where station.name = "MUMBAI");	2 row(s) returned	0.000 sec / 0.000 sec

4. List all the stations in order of visit by the train 'CST-AMR_LOCAL'.

`select trainhalts.stcode from trainhalts inner join train on train.id = trainhalts.id where train.name = 'CST-AMR_LOCAL';`

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

```

91 select train.name from train where train.id in (select trainhalts.id from trainhalts inner join station on trainhalts.stcode = station.stcode where station.name = "MUMBAI");
92 -- 4. List all the stations in order of visit by the train 'CST-AMR_LOCAL'.
93 select trainhalts.stcode from trainhalts inner join train on train.id = trainhalts.id where train.name = 'CST-AMR_LOCAL';
94
95 -- drop table trainhalts;
96 -- drop table train;
97 -- drop table track;
98 -- drop table station;
99
100 -- delete from trainhalts;

```

The left sidebar shows a list of tables: stcode, CST, BYC, DR, KRL, GPR, TNA, DL, KYN, and AMR. The bottom panel shows the 'Action Output' table with the following data:

#	Time	Action	Message	Duration / Fetch
66	06:11:46	select trainhalts.stcode from trainhalts inner join train on train.id = trainhalts.id where train name = 'CST-AMR_LOCAL' LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
67	06:11:52	select train.name from train where train.id in (select trainhalts.id from trainhalts inner join station on trainhalts.stcode = station.stcode ...	2 row(s) returned	0.000 sec / 0.000 sec
68	06:12:28	select trainhalts.stcode from trainhalts inner join train on train.id = trainhalts.id where train name = 'CST-AMR_LOCAL' LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec