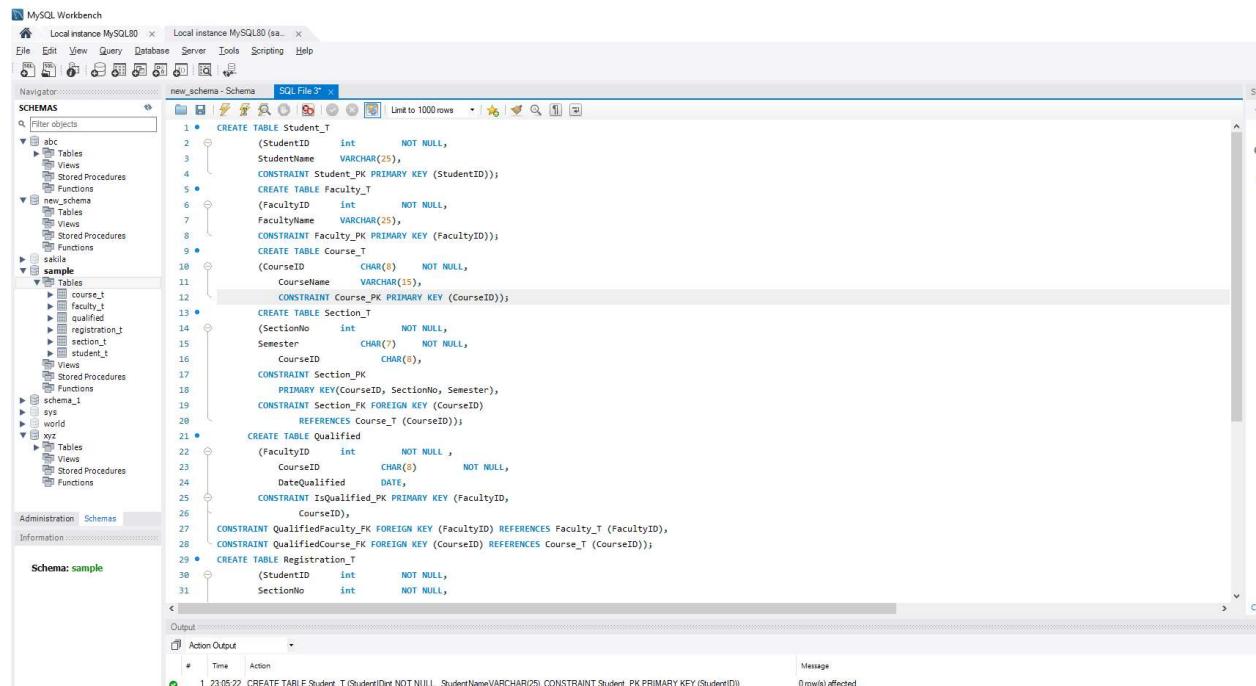


DMDD HOMEWORK 08

Name: Romita Upendra Thally
NUID: 002100103
Email: thally.r@northeastern.edu

Q. 1



The screenshot shows the MySQL Workbench interface with the following details:

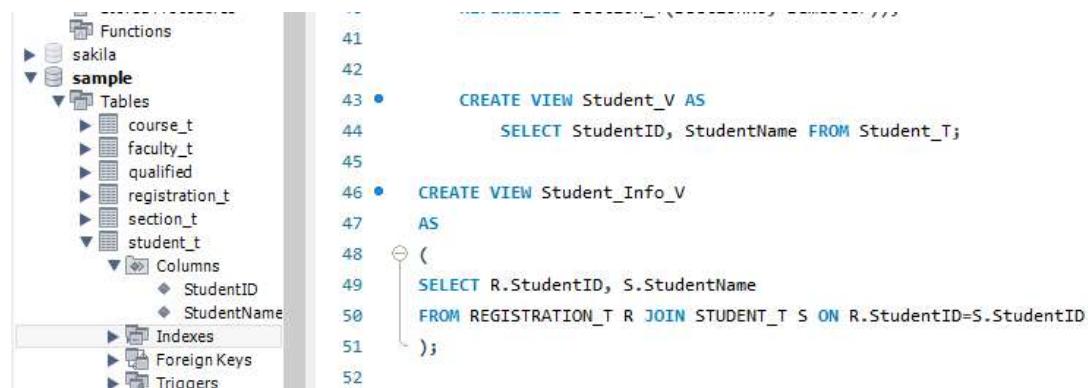
- Schemas:** The current schema is "sample". Other schemas listed include "abc", "new_schema", "sakila", and "xyz".
- Tables:** Several tables are shown in the Navigator pane, including "Student_T", "Faculty_T", "Course_T", "Section_T", "Qualified_T", and "Registration_T".
- SQL Editor:** The main window displays the following SQL code for creating tables and defining constraints:

```
1 • CREATE TABLE Student_T
2   (StudentID      int          NOT NULL,
3    StudentName    VARCHAR(25),
4    CONSTRAINT Student_PK PRIMARY KEY (StudentID));
5 • CREATE TABLE Faculty_T
6   (FacultyID      int          NOT NULL,
7    FacultyName   VARCHAR(25),
8    CONSTRAINT Faculty_PK PRIMARY KEY (FacultyID));
9 • CREATE TABLE Course_T
10  (CourseID       CHAR(8)     NOT NULL,
11   CourseName    VARCHAR(15),
12   CONSTRAINT Course_PK PRIMARY KEY (CourseID));
13 • CREATE TABLE Section_T
14  (SectionNo      int          NOT NULL,
15   Semester       CHAR(?)    NOT NULL,
16   CourseID       CHAR(8),
17   CONSTRAINT Section_PK
18   PRIMARY KEY(CourseID, SectionNo, Semester),
19   CONSTRAINT Section_FK FOREIGN KEY (CourseID)
20   REFERENCES Course_T (CourseID));
21 • CREATE TABLE Qualified
22  (FacultyID      int          NOT NULL ,
23   CourseID       CHAR(8)     NOT NULL,
24   DateQualified  DATE,
25   CONSTRAINT IsQualified_PK PRIMARY KEY (FacultyID,
26   CourseID),
27   CONSTRAINT QualifiedFaculty_FK FOREIGN KEY (FacultyID) REFERENCES Faculty_T (FacultyID),
28   CONSTRAINT QualifiedCourse_FK FOREIGN KEY (CourseID) REFERENCES Course_T (CourseID));
29 • CREATE TABLE Registration_T
30  (StudentID      int          NOT NULL,
31   SectionNo      int          NOT NULL,
```

The output pane shows a single log entry:

```
1 23:05:22 CREATE TABLE Student_T (StudentID INT NOT NULL, StudentName VARCHAR(25), CONSTRAINT Student_PK PRIMARY KEY (StudentID)) 0 row(s) affected
```

Q. 2.



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is "sample". Other schemas listed include "sakila" and "Functions".
- Tables:** The "sample" schema contains tables: "course_t", "faculty_t", "qualified", "registration_t", "section_t", and "student_t".
- Views:** The following SQL code creates two views:

```
41
42
43 • CREATE VIEW Student_V AS
44   SELECT StudentID, StudentName FROM Student_T;
45
46 • CREATE VIEW Student_Info_V
47   AS
48   (
49     SELECT R.StudentID, S.StudentName
50     FROM REGISTRATION_T R JOIN STUDENT_T S ON R.StudentID=S.StudentID
51   );
```

Q. 3.

The screenshot shows the SSMS interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for New Query, Open, Save, Print, Copy, Paste, Find, and Run.
- Navigator:** Shows the database schema tree:
 - SCHEMAS:** abc, new_schema, sakila, sample.
 - sample:** Tables (course_t, faculty_t, qualified, section_t, student_t), Views, Stored Procedures, Functions.
 - student_t:** Columns (StudentID, StudentName, Class), Indexes, Foreign Keys, Triggers.
- Script Editor:** The main window displays the following T-SQL script:

```
53 • CREATE TABLE Section_T
54     (SectionNo      int      NOT NULL,
55      Semester       CHAR(7)   NOT NULL,
56      CourseID      CHAR(8),
57      CONSTRAINT Section_PK
58          PRIMARY KEY(CourseID, SectionNo, Semester),
59      CONSTRAINT Section_FK FOREIGN KEY (CourseID)
60          REFERENCES Course_T (CourseID);

61
62
63
64 • ALTER TABLE Student_T
65     ADD Class VARCHAR(5);
66 • DROP TABLE Registration_T;
67 • ALTER TABLE Faculty_T
68     MODIFY FacultyName VARCHAR(40);
```

Q. 4.

File Edit View Query Database Server Tools Scripting Help

SQL SQL+ Import Export Scripts Reports Data Definition Data Manipulation

Navigator new_schema - Schema SQL File 3* ×

SCHEMAS

Filter objects

abc

- Tables
- Views
- Stored Procedures
- Functions

new_schema

- Tables
- Views
- Stored Procedures
- Functions

sakila

sample

- Tables could not be fetched
 - course_t
 - faculty_t
 - qualified
 - section_t
 - student_t
 - Columns
 - StudentID
 - StudentName
 - Class
 - Indexes
 - Foreign Keys
 - Triggers
- Views could not be fetched

CONSTRAINT Section_FK FOREIGN KEY (CourseID)
REFERENCES Course_T (CourseID);

ALTER TABLE Student_T
ADD Class VARCHAR(5);

DROP TABLE Registration_T;

ALTER TABLE Faculty_T
MODIFY FacultyName VARCHAR(40);

INSERT INTO Student_T (StudentID, StudentName)
VALUES (65798, 'Lopez');

INSERT INTO Student_T VALUES (65798, 'Lopez');

DELETE FROM Student_T WHERE StudentID = 65798;

UPDATE Course_T
SET CourseName = 'DBMS1'
WHERE CourseID = '4212';

Q. 5.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Schemas

Navigator: new_schema - Schema SQL File 3*

```
67 • ALTER TABLE Faculty_T
    MODIFY FacultyName VARCHAR(40);

68
69
70
71 • INSERT INTO STUDENT_T VALUES (12345, 'Romita', NULL);
72 • INSERT INTO STUDENT_T (StudentID, StudentName) VALUES (12345, 'Romita');
73 • DELETE FROM STUDENT_T WHERE StudentID = 12345;
74 • UPDATE COURSE_T SET CourseName = 'DBMS' WHERE CourseID = 'DB 7245';
75
76
77 • select * FROM student_t
78
79
80
81
82
83
84
85
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

StudentID	StudentName	Class
12345	Romita	NULL
65798	Lopez	NULL
*	NULL	NULL

Q. 6.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: new_schema - Schema SQL File 3* x

SCHEMAS

abc

new_schema

sakila

sample

schema_1

sys

world

new_schema - Schema

SQL File 3* x

76

77 • select * FROM student_t;

78

79

80 • SELECT StudentID, StudentName

81 FROM Student_T

82 WHERE StudentID < 50000;

83 • SELECT FacultyName

84 FROM Faculty_T

85 WHERE FacultyID = 4756;

86 • SELECT MIN(SectionID)

87 FROM Registration_T

88 WHERE Semester = 'I-2008';

89

90

91

92

93

94

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

FacultyName

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are two tabs: 'Local instance MySQL80' and 'Local instance MySQL80 (sa...'. The main menu includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The left sidebar is titled 'Navigator' and shows a tree view of databases: 'abc', 'new_schema', 'sakila', and 'sample'. Under 'sample', it lists tables like 'course_t', 'faculty_t', 'qualified', 'section_t', 'student_t', and 'Views could not be fetched'. The 'student_t' table is expanded to show columns: StudentID, StudentName, Class, Indexes, Foreign Keys, and Triggers. The 'Views could not be fetched' section lists 'Stored Procedures could not be found' and 'Functions could not be found'. At the bottom of the sidebar are tabs for 'Administration' and 'Schemas'. The central area is titled 'new_schema - Schema' and contains a 'SQL File 3*' tab. The SQL code in the editor is a complex query involving multiple tables from different schemas. Below the editor is a 'Result Grid' pane with a single column labeled 'FacultyName' containing the value 'FacultyName'. There are also buttons for 'Filter Rows', 'Export', and 'Wrap Cell Content'.

Q. 7

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: new_schema - Schema SQL File 3*

```

81     FROM Student_T
82     WHERE StudentID < 50000;
83 •   SELECT FacultyName
84     FROM Faculty_T
85     WHERE FacultyID = 4756;
86 •   SELECT MIN(SectionID)
87     FROM Registration_T
88     WHERE Semester = 'I-2008';
89
90
91 •   SELECT COUNT(*)
92     FROM Registration_T
93     WHERE Section = 3456
94     AND Semester = 'I-2020';
95 •   SELECT FacultyID,CourseID,DateQualified
96     FROM Qualified_T
97     WHERE DateQualified >= '01-JAN-1996';
98
99
100

```

Q. 8.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: new_schema - Schema SQL File 3*

```

102     WHERE SectionNo IN (2714,2715)
103     GROUP BY StudentID
104     HAVING COUNT(*) > 1;
105
106
107
108
109
110
111
112 •   SELECT SC.StudentID, ST.StudentName
113     FROM
114     (
115       SELECT RT.StudentID,
116             COUNT(*) Sub_Count
117           FROM REGISTRATION_RT
118         JOIN SECTION_ST ON RT.SectionNo = ST.SectionNo AND RT.Semester = ST.Semester
119         JOIN COURSE_CT ON ST.CourseID = CT.CourseID
120         WHERE CT.CourseName IN ('Networking','Introduction to Relational Databases') GROUP BY RT.StudentID
121     ) SC JOIN STUDENT_SI ON SC.StudentID = SI.StudentID WHERE SC.Sub_Count > 1;
122 •   SELECT DISTINCT QT.FacultyID, FT.FacultyName
123     FROM QUALIFIED_T QT
124     JOIN COURSE_T CT ON QT.CourseID = CT.CourseID
125     JOIN FACULTY_FT ON QT.FacultyID = FT.FacultyID WHERE CourseName NOT IN ('Syst Analysis','Syst Design')
126
127
128
129

```