

# COSC 4606 Project 2

## Report

Darrel Asare

### Screenshots for Functional Requirements


Testing java\_code for database administrators using root.

```
MyJDBC.java
1  import java.sql.Connection;
2  import java.sql.DriverManager;
3  import java.sql.ResultSet;
4  import java.sql.Statement;
5  |
6  public class MyJDBC {
7  public static void main(String args[]){
8      try{
9          Class.forName("com.mysql.jdbc.Driver");
10         Connection con= DriverManager.getConnection(
11             url: "jdbc:mysql://localhost:3306/academic_records", user: "root", password: "RootAdmin@");
12         //here academic_records is database name, root is the username and Decagypysu@9k is the password
13         Statement stmt=con.createStatement();
14         ResultSet rs=stmt.executeQuery( sql: "select * from academic_staff");
15         while(rs.next())
16             System.out.println(rs.getInt( columnIndex: 1)+" " +rs.getString( columnIndex: 2)+" "
17                 +rs.getString( columnIndex: 3)+" " +rs.getString( columnIndex: 4)+" "
18                 +rs.getDate( columnIndex: 5)+" " +rs.getDate( columnIndex: 6)+" "
19                 +rs.getString( columnIndex: 7)+" " +rs.getString( columnIndex: 8)+" "
20                 +rs.getString( columnIndex: 9)+" " +rs.getDate( columnIndex: 10)+" "
21                 +rs.getString( columnIndex: 11)+" " +rs.getString( columnIndex: 12)+" "
22                 +rs.getString( columnIndex: 13));
23         con.close();
24     }catch(Exception e){ System.out.println(e);}
25 }
26 }
```

### Academic\_staff

```
231 Kakashi Hatake Kakashi@mail.com 2021-09-07 2021-12-18 1958 R S Regina 306-533-7822 1980-03-12 3506 Fall 3706
232 Minato Namikaze Minato@mail.com 2022-01-12 2022-04-26 563 F D Neustadt 519-799-6367 1984-05-04 3707 Winter 2707
233 John Zao John@mail.com 2021-09-07 2021-12-18 1819 D P Coniston 705-694-0503 1986-11-15 3596 Fall 3706
234 Ryu Effron ryu@mail.com 2022-01-12 2022-04-26 193 C H Montreal 514-265-8268 1981-02-20 2056 Winter null
235 Floyd Lawton floyed@mail.com 2021-09-07 2021-12-18 29 L A Brantford 519-755-8613 1978-03-11 3506 Fall null
236 Oliver King oliver@mail.com 2022-01-12 2022-04-26 3992 A Edmonton 780-604-6789 1990-06-16 2707 Winter null
```

### Registrar privileges



COSC4606

## Users and Privileges

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
registrar	%
root	localhost

Details for account registrar@%

Login

Account Limits

Administrative Roles

Schema Privileges

Schema	Privileges
academic_records	INSERT, SELECT, UPDATE

<

>

Schema and Host fields may use % and \_ wildcards.  
The server will match specific entries before wildcarded ones.

Revoke All Privileges

Delete Entry

Add Entry...

The user 'registrar'@'%' will have the following access rights to schemas matching 'academic\_records':

Object Rights

☒ SELECT  
☒ INSERT  
☒ UPDATE  
☐ DELETE  
☐ EXECUTE  
☐ SHOW VIEW

DDL Rights

☐ CREATE  
☐ ALTER  
☐ REFERENCES  
☐ INDEX  
☐ CREATE VIEW  
☐ CREATE ROUTINE  
☐ ALTER ROUTINE  
☐ EVENT  
☐ DROP

Other Rights

☐ GRANT OPTION  
☐ CREATE TEMPORARY TABLES  
☐ LOCK TABLES

## Original students table

	StudentID	FirstName	LastName	Email	DateOfBirth	Address	City	MobilePhone	Course_Enroll_ID	StartDate	EndDate
▶	121	Ichigo	Kuro	ichigo@mail.com	1998-09-24 00:00:00	2864 Rd	San Jose	408-363-8840	3506	2021-09-08 00:00:00	2021-12-17 00:00:00
	122	Boruto	Uzumaki	boruto@mail.com	2001-03-22 00:00:00	4041 L A	New York	718-572-2644	3707	2022-01-13 00:00:00	2022-04-25 00:00:00
	123	Renji	Abari	renji@mail.com	1998-02-17 00:00:00	1556 V P	Kinuso	780-775-8665	3706	2021-09-08 00:00:00	2021-12-17 00:00:00
	124	Matt	Murdock	matt@mail.com	2000-06-16 00:00:00	911 K Rd	North Vancouver	604-981-8242	2707	2022-01-13 00:00:00	2022-04-25 00:00:00
	125	Shawn	Rollins	shawn@mail.com	1999-07-09 00:00:00	1299 I A	Toronto	416-259-7500	3596	2021-09-08 00:00:00	2021-12-17 00:00:00
	126	Seth	Rogan	seth@mail.com	2000-01-12 00:00:00	2626 P B	Dorval	514-403-3403	2056	2022-01-13 00:00:00	2022-04-25 00:00:00

## Testing java code for registrar inserting values

```

1  import java.sql.Connection;
2  import java.sql.DriverManager;
3  import java.sql.ResultSet;
4  import java.sql.Statement;
5
6  public class JDBCRegistrar {
7      public static void main(String args[]){
8          try{
9              Class.forName("com.mysql.jdbc.Driver");
10             Connection con= DriverManager.getConnection(
11                 url: "jdbc:mysql://localhost:3306/academic_records", user: "registrar", password: "Registrar@");
12
13             Statement stmt=con.createStatement();
14             // Execute SQL Query
15             String sql = "insert into students"
16                 + " (StudentID, FirstName, LastName, Email, DateOfBirth, " +
17                 "Address, City, MobilePhone, Course_Enroll_ID, StartDate, EndDate)"
18                 + " values ('127', 'Brock', 'Rain', 'brock@mail.com', '1998-02-26', " +
19                 "'1565 FD', 'Harrow', '519-738-5371', '3506', '2021-09-08', '2021-12-17')";
20
21             stmt.executeUpdate(sql);
22             System.out.println("Insert complete");
23             con.close();
24         }catch(Exception e){ System.out.println(e);}
25     }
26 }

```

Insert complete

Process finished with exit code 0

### Students table after insert

	StudentID	FirstName	LastName	Email	DateOfBirth	Address	City	MobilePhone	Course_Enroll_ID	StartDate	EndDate
▶	121	Ichigo	Kuro	ichigo@mail.com	1998-09-24 00:00:00	2864 Rd	San Jose	408-363-8840	3506	2021-09-08 00:00:00	2021-12-17 00:00:00
	122	Boruto	Uzumaki	boruto@mail.com	2001-03-22 00:00:00	4041 L A	New York	718-572-2644	3707	2022-01-13 00:00:00	2022-04-25 00:00:00
	123	Renji	Abari	renji@mail.com	1998-02-17 00:00:00	1556 V P	Kinuso	780-775-8665	3706	2021-09-08 00:00:00	2021-12-17 00:00:00
	124	Matt	Murdock	matt@mail.com	2000-06-16 00:00:00	911 K Rd	North Vancouver	604-981-8242	2707	2022-01-13 00:00:00	2022-04-25 00:00:00
	125	Shawn	Rollins	shawn@mail.com	1999-07-09 00:00:00	1299 I A	Toronto	416-259-7500	3596	2021-09-08 00:00:00	2021-12-17 00:00:00
	126	Seth	Rogan	seth@mail.com	2000-01-12 00:00:00	2626 P B	Dorval	514-403-3403	2056	2022-01-13 00:00:00	2022-04-25 00:00:00
	127	Brock	Rain	brock@mail.com	1998-02-26 00:00:00	1565 FD	Harrow	519-738-5371	3506	2021-09-08 00:00:00	2021-12-17 00:00:00

## Java code for Registrar update to students table

```
public class JDBCRegistrar {  
    public static void main(String args[]){  
        try{  
            Class.forName("com.mysql.jdbc.Driver");  
            Connection con= DriverManager.getConnection(  
                url: "jdbc:mysql://localhost:3306/academic_records", user: "registrar", password: "Registrar@");  
  
            Statement stmt=con.createStatement();  
            // Execute SQL Query  
            /* String sql = "insert into students"  
                + " (StudentID, FirstName, LastName, Email, DateOfBirth, " +  
                "Address, City, MobilePhone, Course_Enroll_ID, StartDate, EndDate)"  
                + " values ('127', 'Brock', 'Rain', 'brock@mail.com', '1998-02-26', " +  
                "'1565 FD', 'Harrow', '519-738-5371', '3506', '2021-09-08', '2021-12-17')"; */  
  
            // Execute SQL Query  
            String sql = "update students "  
                + " set Email = 'brock@gmail.com'"  
                + " where StudentID = 127";  
  
            stmt.executeUpdate(sql);  
            // System.out.println("Insert complete");  
            System.out.println("Update complete");  
            con.close();  
        }catch(Exception e){ System.out.println(e);}  
    }  
}
```

Update complete

Process finished with exit code 0

Students table after updating email to [brock@gmail.com](mailto:brock@gmail.com) instead of brock@mail.com

	StudentID	FirstName	LastName	Email	DateOfBirth	Address	City	MobilePhone	Course_Enroll_ID	StartDate	EndDate
▶	121	Ichigo	Kuro	ichigo@mail.com	1998-09-24 00:00:00	2864 Rd	San Jose	408-363-8840	3506	2021-09-08 00:00:00	2021-12-17 00:00:00
	122	Boruto	Uzumaki	boruto@mail.com	2001-03-22 00:00:00	4041 L A	New York	718-572-2644	3707	2022-01-13 00:00:00	2022-04-25 00:00:00
	123	Renji	Abari	renji@mail.com	1998-02-17 00:00:00	1556 V P	Kinuso	780-775-8665	3706	2021-09-08 00:00:00	2021-12-17 00:00:00
	124	Matt	Murdock	matt@mail.com	2000-06-16 00:00:00	911 K Rd	North Vancouver	604-981-8242	2707	2022-01-13 00:00:00	2022-04-25 00:00:00
	125	Shawn	Rollins	shawn@mail.com	1999-07-09 00:00:00	1299 I A	Toronto	416-259-7500	3596	2021-09-08 00:00:00	2021-12-17 00:00:00
	126	Seth	Rogan	seth@mail.com	2000-01-12 00:00:00	2626 P B	Dorval	514-403-3403	2056	2022-01-13 00:00:00	2022-04-25 00:00:00
	127	Brock	Rain	brock@gmail.com	1998-02-26 00:00:00	1565 FD	Harrow	519-738-5371	3506	2021-09-08 00:00:00	2021-12-17 00:00:00

project2\* Administration - Users and Privileges x

COSC4606  
Users and Privileges

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
registrar	%
root	localhost
student	%

Details for account student@%

Login Account Limits Administrative Roles Schema Privileges

Schema	Privileges
academic_records	SELECT

Java code for student to check academic summary

```
JDBCStudent.java x
1 // Darrel Asare
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.ResultSet;
5 import java.sql.Statement;
6
7 public class JDBCStudent {
8     public static void main(String args[]){
9         try{
10             Class.forName("com.mysql.jdbc.Driver");
11             Connection con= DriverManager.getConnection(
12                 "jdbc:mysql://localhost:3306/academic_records", user: "student", password: "Student@");
13
14             Statement stmt=con.createStatement();
15             // Execute SQL Query
16             ResultSet rs=stmt.executeQuery("select * from academic_summary limit 1");
17             while(rs.next())
18                 System.out.println(rs.getInt( columnIndex: 1)+" " +rs.getInt( columnIndex: 2)+" "
19                     +rs.getInt( columnIndex: 3)+" " +rs.getInt( columnIndex: 4)+" "
20                     +rs.getInt( columnIndex: 5)+" " +rs.getString( columnIndex: 6)+" "
21                     +rs.getString( columnIndex: 7));
22             con.close();
23         }catch(Exception e){ System.out.println(e);}
24     }
25 }
```

```
121 2021 75 79 3707 Systems Analysis Winter
Process finished with exit code 0
```


Academic\_summary table view

```
mysql> select * from academic_summary;
```

StudentID	Year	Final_Grades	GPA	CourseID	Course_Names	Term
121	2021	75	79	3707	Systems Analysis	Winter
121	2021	75	79	3706	Project Management	Fall
121	2021	75	79	3596	Mobile Development	Fall
121	2021	75	79	3506	Software Engineering	Fall
121	2021	75	79	2707	E-Business Architect	Winter
121	2021	75	79	2056	Discrete Math	Winter
125	2021	90	82	3707	Systems Analysis	Winter
125	2021	90	82	3706	Project Management	Fall
125	2021	90	82	3596	Mobile Development	Fall
125	2021	90	82	3506	Software Engineering	Fall
125	2021	90	82	2707	E-Business Architect	Winter
125	2021	90	82	2056	Discrete Math	Winter
122	2022	70	77	3707	Systems Analysis	Winter
122	2022	70	77	3706	Project Management	Fall
122	2022	70	77	3596	Mobile Development	Fall
122	2022	70	77	3506	Software Engineering	Fall
122	2022	70	77	2707	E-Business Architect	Winter
122	2022	70	77	2056	Discrete Math	Winter
126	2022	68	72	3707	Systems Analysis	Winter
126	2022	68	72	3706	Project Management	Fall
126	2022	68	72	3596	Mobile Development	Fall
126	2022	68	72	3506	Software Engineering	Fall
126	2022	68	72	2707	E-Business Architect	Winter
126	2022	68	72	2056	Discrete Math	Winter
123	2021	74	78	3707	Systems Analysis	Winter
123	2021	74	78	3706	Project Management	Fall
123	2021	74	78	3596	Mobile Development	Fall
123	2021	74	78	3506	Software Engineering	Fall
123	2021	74	78	2707	E-Business Architect	Winter
123	2021	74	78	2056	Discrete Math	Winter
124	2022	88	80	3707	Systems Analysis	Winter
124	2022	88	80	3706	Project Management	Fall
124	2022	88	80	3596	Mobile Development	Fall
124	2022	88	80	3506	Software Engineering	Fall
124	2022	88	80	2707	E-Business Architect	Winter
124	2022	88	80	2056	Discrete Math	Winter

project2

Administration - Users and Privil...



COSC4606

Users and Privileges

User Accounts

User	From Host
faculty	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
registrar	%
root	localhost
student	%

Details for account faculty@%

Login

Account Limits

Administrative Roles

Schema Privileges

Schema	Privileges
academic_records	SELECT

Schema and Host fields may use % and \_ wildcards.

The server will match specific entries before wildcarded ones.

The user 'faculty'@'%' will have the following access rights to schemas matching 'academic\_records':

Object Rights

☒ SELECT  
☐ INSERT  
☐ UPDATE  
☐ DELETE  
☐ EXECUTE  
☐ SHOW VIEW

DOL Rights

☐ CREATE  
☐ ALTER  
☐ REFERENCES  
☐ INDEX  
☐ CREATE VIEW  
☐ CREATE ROUTINE  
☐ ALTER ROUTINE  
☐ EVENT  
☐ DROP

Other Rights

☐ GRANT OPTION  
☐ CREATE TEMPORARY TABLES  
☐ LOCK TABLES

Revoke All Privileges

Delete Entry

Add Entry...

Java code for faculty to check course list he is teaching

```
JDBCFaculty.java x
1 // Darrel Asare
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7
8 public class JDBCFaculty {
9     public static void main(String args[]){
10         try{
11             Class.forName("com.mysql.jdbc.Driver");
12             Connection con= DriverManager.getConnection(
13                 url: "jdbc:mysql://localhost:3306/academic_records", user: "faculty", password: "Faculty@");
14
15             Statement stmt=con.createStatement();
16             // Execute SQL Query
17             ResultSet rs=stmt.executeQuery( sql: "select * from course_list limit 1");
18             while(rs.next())
19                 System.out.println(rs.getInt( columnIndex: 1)+" "+rs.getString( columnIndex: 2)+" "
20                                     +rs.getString( columnIndex: 3)+" " +rs.getInt( columnIndex: 4)+" "
21                                     +rs.getString( columnIndex: 5)+" " +rs.getString( columnIndex: 6)+" "
22                                     +rs.getString( columnIndex: 7));
23             con.close();
24         }catch(Exception e){ System.out.println(e);}
25     }
26 }
```

```
2056 Discrete Math Winter 5 Mr.Effron 234 0

Process finished with exit code 0
```

Course\_list table view

	CourseID	Course_Names	Term	sections	Faculty_Teaching_Course	FacultyID	Team_Teaching
▶	2056	Discrete Math	Winter	5	Mr.Effron	234	0
	2707	E-Business Architect	Winter	8	Mr.Namikaze	232	0
	3506	Software Engineering	Fall	2	Mr.Hatake	231	1
	3596	Mobile Development	Fall	1	Mr.Zao	233	1
	3706	Project Management	Fall	10	Mr.Lawton	235	0
	3707	Systems Analysis	Winter	4	Mr.Namikaze	232	1

## Forms, Reports and Deliverables

### User Manual

This report will explain the project of an academic record system using pictures and paragraphs. This time the tables and forms will be shown from SQL Workbench working with SQL Server.

### Evidence of normalized tables

All the tables are in 1NF because each attribute in a relation has atomic single values. In the Co-op table, the student name attribute also has the student first names underscore last names. We also have a student course deletion table which is in 2NF because we moved fields from the courses table which means they do not have to be fully dependent on the primary key. This makes it easier to see which student has deleted a course instead of adding that information to the courses table, which will cause the table to increase in size.

### Printouts of test runs for forms and reports with table and query structures

#### Forms

#### 1. add a faculty member to the academic staff

	FacultyID	FirstName	LastName	Email	StartDate	EndDate	Address	City	MobilePhone	DateOfBirth	First_Course_Offered	Term	Second_Course_Offered
▶	231	Kakashi	Hatake	Kakashi@mail.com	2021-09-07 00:00:00	2021-12-18 00:00:00	1958 R S	Regina	306-533-7822	1980-03-12 00:00:00	3506	Fall	3706
	232	Minato	Namikaze	Minato@mail.com	2022-01-12 00:00:00	2022-04-26 00:00:00	563 F D	Neustadt	519-799-6367	1984-05-04 00:00:00	3707	Winter	2707
	233	John	Zao	John@mail.com	2021-09-07 00:00:00	2021-12-18 00:00:00	1819 D P	Coriston	705-694-0503	1986-11-15 00:00:00	3596	Fall	3706
	234	Ryu	Effron	ryu@mail.com	2022-01-12 00:00:00	2022-04-26 00:00:00	193 C H	Montreal	514-265-8268	1981-02-20 00:00:00	2056	Winter	NULL
	235	Floyd	Lawton	floyd@mail.com	2021-09-07 00:00:00	2021-12-18 00:00:00	29 L A	Brantford	519-755-8613	1978-03-11 00:00:00	3506	Fall	NULL
	236	Oliver	King	oliver@mail.com	2022-01-12 00:00:00	2022-04-26 00:00:00	3992 A	Edmonton	780-604-6789	1990-06-16 00:00:00	2707	Winter	NULL



Form Editor | Navigate: 1 / 6

FacultyID: 231

FirstName: Kakashi

LastName: Hatake

Email: Kakashi@mail.com

StartDate: 2021-09-07 00:00:00

EndDate: 2021-12-18 00:00:00

Address: 1958 R S

City: Regina

MobilePhone: 306-533-7822

DateOfBirth: 1980-03-12 00:00:00

First\_Course\_Offered: 3506

Term: Fall

Second\_Course\_Offered: 3706

## 2. register a student in the university

	StudentID	FirstName	LastName	Email	DateOfBirth	Address	City	MobilePhone	Course_Enroll_ID	StartDate	EndDate
▶	121	Ichigo	Kuro	ichigo@mail.com	1998-09-24 00:00:00	2864 Rd	San Jose	408-363-8840	3506	2021-09-08 00:00:00	2021-12-17 00:00:00
	122	Boruto	Uzumaki	boruto@mail.com	2001-03-22 00:00:00	4041 L A	New York	718-572-2644	3707	2022-01-13 00:00:00	2022-04-25 00:00:00
	123	Renji	Abari	renji@mail.com	1998-02-17 00:00:00	1556 V P	Kinuso	780-775-8665	3706	2021-09-08 00:00:00	2021-12-17 00:00:00
	124	Matt	Murdock	matt@mail.com	2000-06-16 00:00:00	911 K Rd	North Vancouver	604-981-8242	2707	2022-01-13 00:00:00	2022-04-25 00:00:00
	125	Shawn	Rollins	shawn@mail.com	1999-07-09 00:00:00	1299 I A	Toronto	416-259-7500	3596	2021-09-08 00:00:00	2021-12-17 00:00:00
	126	Seth	Rogan	seth@mail.com	2000-01-12 00:00:00	2626 P B	Dorval	514-403-3403	2056	2022-01-13 00:00:00	2022-04-25 00:00:00
	127	Brock	Rain	brock@gmail.com	1998-02-26 00:00:00	1565 FD	Harrow	519-738-5371	3506	2021-09-08 00:00:00	2021-12-17 00:00:00

Form Editor | Navigate: ⏮ ⏪ 1 / 7 ⏩ ⏭

StudentID: 121

FirstName: Ichigo

LastName: Kuro

Email: ichigo@mail.com

DateOfBirth: 1998-09-24 00:00:00

Address: 2864 Rd

City: San Jose

MobilePhone: 408-363-8840

Course\_Enroll\_ID: 3506

StartDate: 2021-09-08 00:00:00

EndDate: 2021-12-17 00:00:00

- assign a faculty member to teach a given section of a given course in a given term, permitting the possibility of “team teaching” with more than one faculty member for a given section

	CourseID	Course_Names	Term	sections	Faculty_Teaching_Course	FacultyID	Team_Teaching
▶	2056	Discrete Math	Winter	5	Mr.Effron	234	0
	2707	E-Business Architect	Winter	8	Mr.Namikaze	232	0
	3506	Software Engineering	Fall	2	Mr.Hatake	231	1
	3596	Mobile Development	Fall	1	Mr.Zao	233	1
	3706	Project Management	Fall	10	Mr.Lawton	235	0
	3707	Systems Analysis	Winter	4	Mr.Namikaze	232	1

Form Editor | Navigate: 1 / 6

CourseID: 2056

Course\_Names: Discrete Math

Term: Winter

Sections: 5

Faculty\_Teaching\_Course: Mr.Effron

FacultyID: 234

Team\_Teaching: 0

4. enroll a student in a particular course  
Shown in the students table with course enroll id.

5. delete a student from a particular course

Result Grid | Filter Rows: | Export:

	DeletionID	StudentID	CourseID	Course_Name
▶	301	121	3707	Systems Analysis
	302	122	2056	Discrete Math
	303	123	0	NULL
	304	124	0	NULL
	305	125	0	NULL
	306	126	0	NULL

Form Editor | Navigate: 1 / 6

DeletionID: 301

StudentID: 121

CourseID: 3707

Course\_Name: Systems Analysis

- alter a student record to permit, for example, entering a grade for a completed course

	GradeID	Final_Grades	Course_Name	Faculty_Name	StudentID	GPA	Year	Term
▶	401	75	Software Engineering	Mr.Hatake	121	79	2021	Fall
	402	90	Mobile Development	Mr.Zao	125	82	2021	Fall
	403	70	Systems Analysis	Mr.Namikaze	122	77	2022	Winter
	404	68	Discrete Math	Mr.Effron	126	72	2022	Winter
	405	74	Project Management	Mr.Zao	123	78	2021	Fall
	406	88	E-Business Architect	Mr.Namikaze	124	80	2022	Winter

Form Editor | Navigate: 1/6

GradeID:

401

Final\_Grades:

75

Course\_Name:

Software Engineering

Faculty\_Name:

Mr.Hatake

StudentID:

121

GPA:

79

Year:

2021

Term:

Fall

## Reports and Information from project 1

- display a class list, given a course id, term and section id, sorted by student name. Indicate whether each student is full-time or part-time that term and whether the student is in the co-op program.

## Co-op Information

StudentName	CourseID	Courses_Term	sections	Co-op_ID	StudentID	Full_Time	Part_Time	Co-op_Program
Boruto_Uzumaki	3707	Winter	4	642	122	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ichigo_Kuro	3506	Fall	2	641	121	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Matt_Murdock	2707	Winter	8	644	124	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Renji_Abari	3706	Fall	10	643	123	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Seth_Rogan	2056	Winter	5	646	126	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Shawn_Rollins	3596	Fall	1	645	125	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

October 11, 2021

Page 1 of 1

- display a student transcript, given a student id. Show all courses enrolled, including failed courses and dropped courses, with an average grade displayed for each academic term and the average overall, using in the latter case only the best mark for repeated courses. The transcript should be ordered by academic year, earliest to latest, and within a given year by course id.

## Student Transcript

Year	Final_Grades	DeletionID	Student_Course_Deletion	Repeated_Course_Name	StudentID	GPA	Term
2022	68	302	Discrete Math	Discrete Math	126	72	Winter
2022	70	301	Systems Analysis	Systems Analysis	122	77	Winter

Final grades after they repeated the deleted course.

- display a table of all students (name, id and contact phone number) currently enrolled in a given degree program (eg. B.Sc.4

COSC), sorted by student name. (Provide sufficient data to be able to run several tests)

Degree Program					
StudentName	ProgramID	StudentID	BSC	COSC	MobileNumber
Boruto_Uzumaki	722	122	<input checked="" type="checkbox"/>	<input type="checkbox"/>	718-572-2644
Ichigo_Kuro	721	121	<input type="checkbox"/>	<input checked="" type="checkbox"/>	408-363-8840
Matt_Murdock	724	124	<input type="checkbox"/>	<input checked="" type="checkbox"/>	604-981-8242
Renji_Abari	723	123	<input checked="" type="checkbox"/>	<input type="checkbox"/>	780-775-8665
Seth_Rogan	726	126	<input type="checkbox"/>	<input checked="" type="checkbox"/>	514-403-3403
Shawn_Rollins	725	125	<input checked="" type="checkbox"/>	<input type="checkbox"/>	416-259-7500

October 11, 2021 Page 1 of 1

4. display the instructor name, course id and home phone number of all instructors teaching a course to some student in a given term, given the student id and term.

Teaching Course Info				
Faculty_Teaching_Course	CourseID	MobilePhone	Term	StudentID
Mr.Effron	2056	514-265-8268	Winter	<input type="text" value="126"/>
Mr.Namikaze	2707	519-799-6367	Winter	<input type="text" value="124"/>
Mr.Hatake	3506	306-533-7822	Fall	<input type="text" value="121"/>
Mr.Zao	3596	705-694-0503	Fall	<input type="text" value="125"/>
Mr.Lawton	3706	519-755-8613	Fall	<input type="text" value="123"/>
Mr.Namikaze	3707	519-799-6367	Winter	<input type="text" value="122"/>

October 12, 2021 Page 1 of 1

5. display a list of all courses taught by a faculty member, grouped by term (earliest to latest) and sorted by course id, given an instructor id and either a term, a range of

List Courses Taught				
Term	CourseID	Course_Names	FacultyID	Faculty_Teaching_Course
Fall	3506	Software Engineering	231	Mr.Hatake
	3596	Mobile Development	233	Mr.Zao
	3706	Project Management	235	Mr.Lawton
Winter	2056	Discrete Math	234	Mr.Effron
	2707	E-Business Architect	232	Mr.Namikaze
	3707	Systems Analysis	232	Mr.Namikaze

October 12, 2021 Page 1 of 1

## Database design for ER diagrams and database schemas

