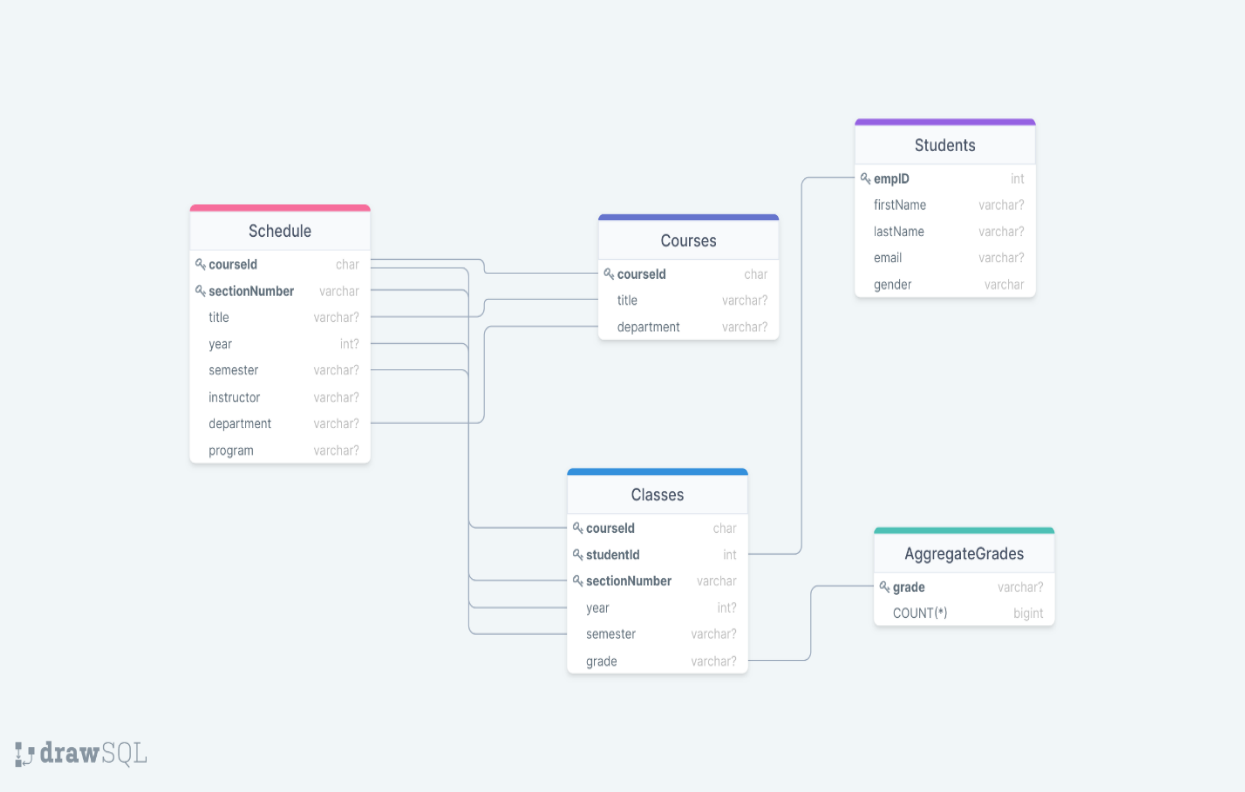
JavaFX Project Part 4

Cristian Statescu

Part 1: Code (look within the repository for the code)

Part 2:
ER DIAGRAM



NOTE: primary keys are marked with a key symbol, and types of the columns are also marked (varchar for example). Foreign keys are shown by the connecting lines.

For reference (in case unclear):

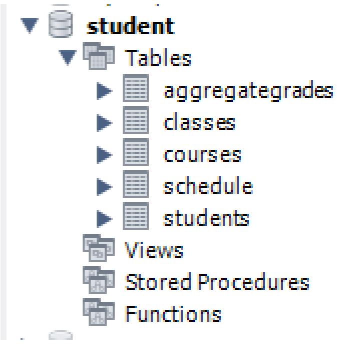
Courses is made from Schedule
Classes is made from Schedule and Students
AggregateGrades is made from classes
Schedule and Students are tables made independently

CONSTRAINTS:

GRADES: ONLY ‘A’, ‘B’, ‘C’, ‘D’, ‘F’, OR ‘W’
GENDER: ONLY ‘M’, ‘F’, OR ‘U’

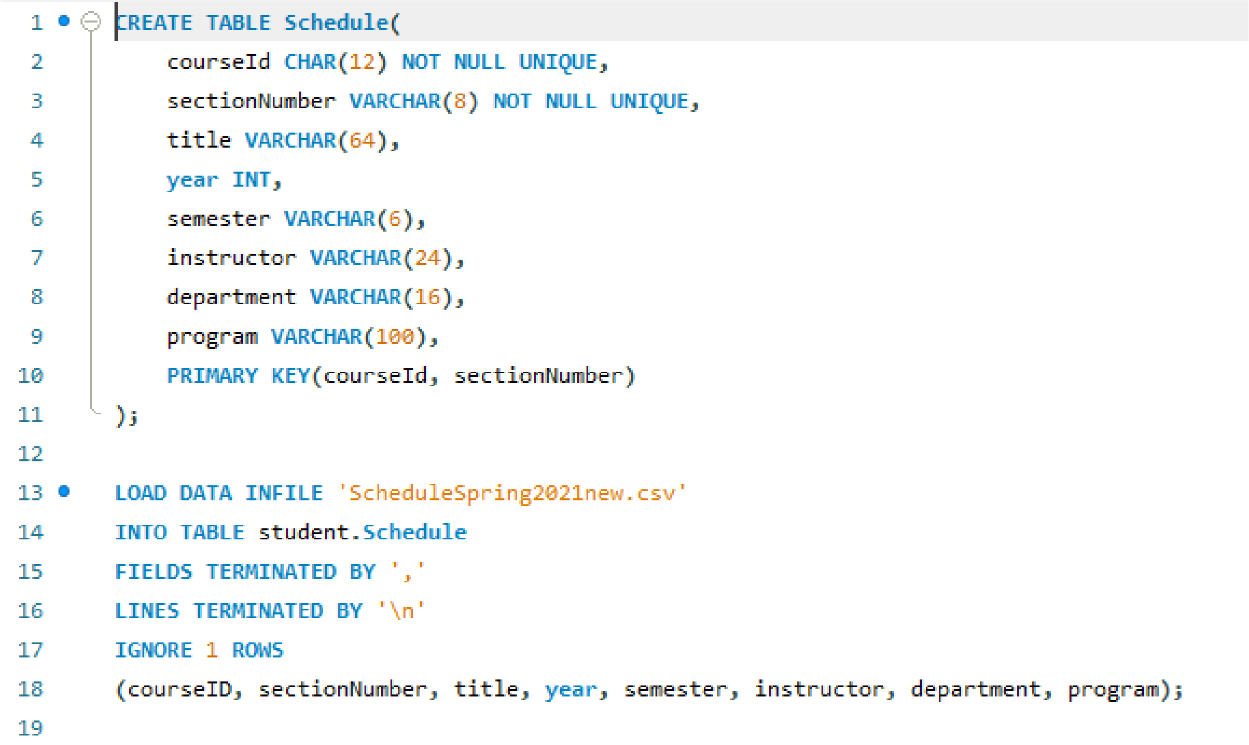
Schema

(database is named ‘student’)



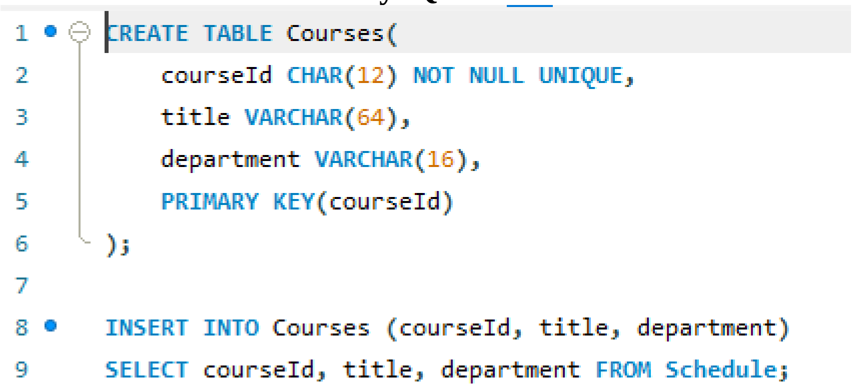
Part 3:

Schedule table creation in MySQL



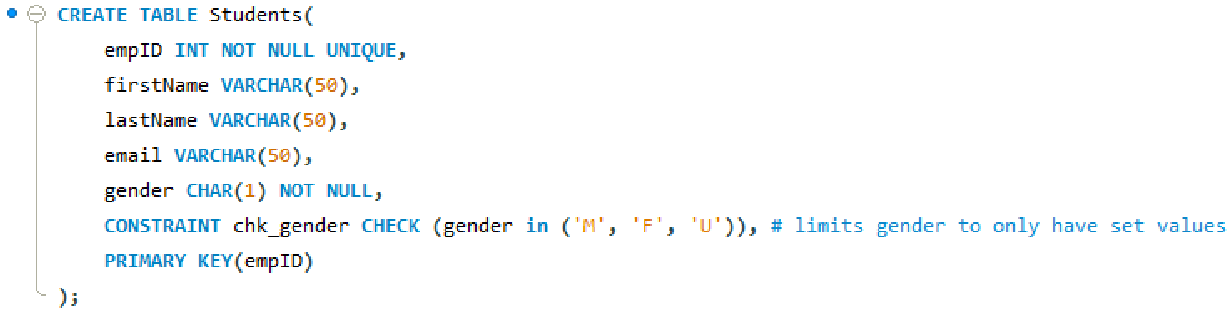
```
CREATE TABLE Schedule(
    courseId CHAR(12) NOT NULL UNIQUE,
    sectionNumber VARCHAR(8) NOT NULL UNIQUE,
    title VARCHAR(64),
    year INT,
    semester VARCHAR(6),
    instructor VARCHAR(24),
    department VARCHAR(16),
    program VARCHAR(100),
    PRIMARY KEY(courseId, sectionNumber)
);

LOAD DATA INFILE 'ScheduleSpring2021new.csv'
INTO TABLE student.Schedule
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(courseID, sectionNumber, title, year, semester, instructor, department, program);
```

Courses table creation in MySQL



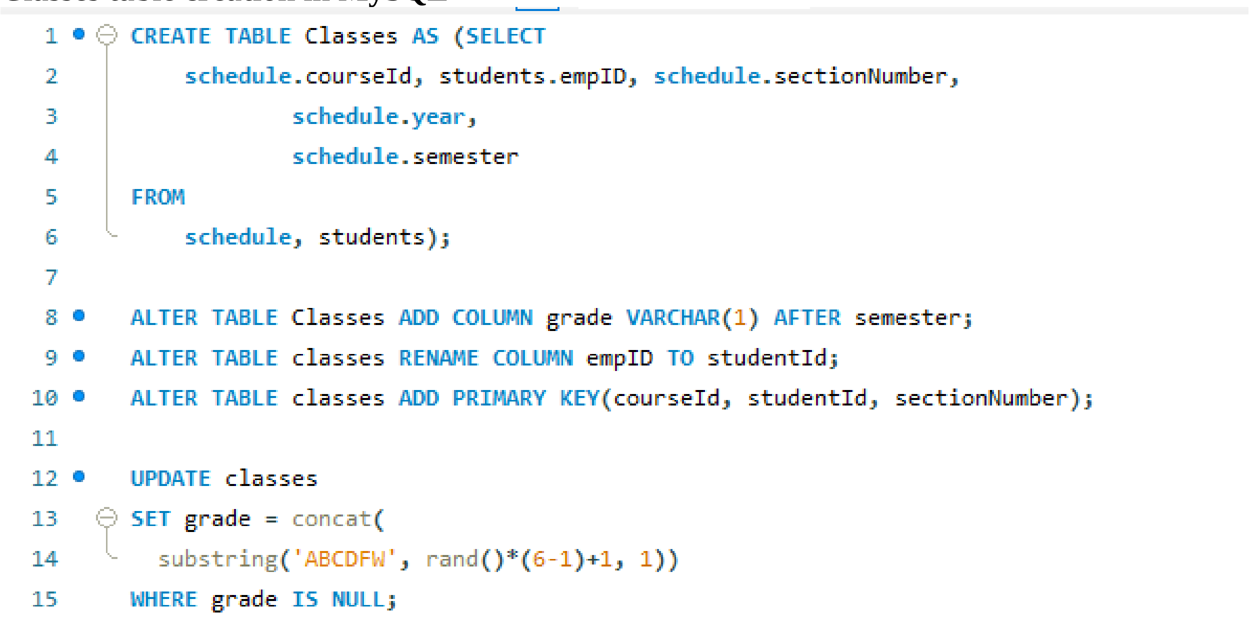
```
CREATE TABLE Courses(
    courseId CHAR(12) NOT NULL UNIQUE,
    title VARCHAR(64),
    department VARCHAR(16),
    PRIMARY KEY(courseId)
);

INSERT INTO Courses (courseId, title, department)
SELECT courseId, title, department FROM Schedule;
```

Students table creation in MySQL



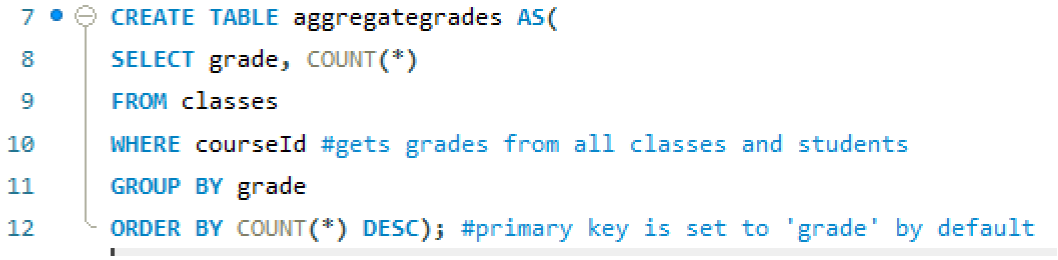
```
CREATE TABLE Students(
    empID INT NOT NULL UNIQUE,
    firstName VARCHAR(50),
    lastName VARCHAR(50),
    email VARCHAR(50),
    gender CHAR(1) NOT NULL,
    CONSTRAINT chk_gender CHECK (gender in ('M', 'F', 'U')), # limits gender to only have set values
    PRIMARY KEY(empID)
);
```

Classes table creation in MySQL



```
CREATE TABLE Classes AS (SELECT
    schedule.courseId, students.empID, schedule.sectionNumber,
            schedule.year,
            schedule.semester
FROM
    schedule, students);

ALTER TABLE Classes ADD COLUMN grade VARCHAR(1) AFTER semester;
ALTER TABLE classes RENAME COLUMN empID TO studentId;
ALTER TABLE classes ADD PRIMARY KEY(courseId, studentId, sectionNumber);

UPDATE classes
SET grade = concat(
  substring('ABCDFW', rand()*(6-1)+1, 1))
WHERE grade IS NULL;
```

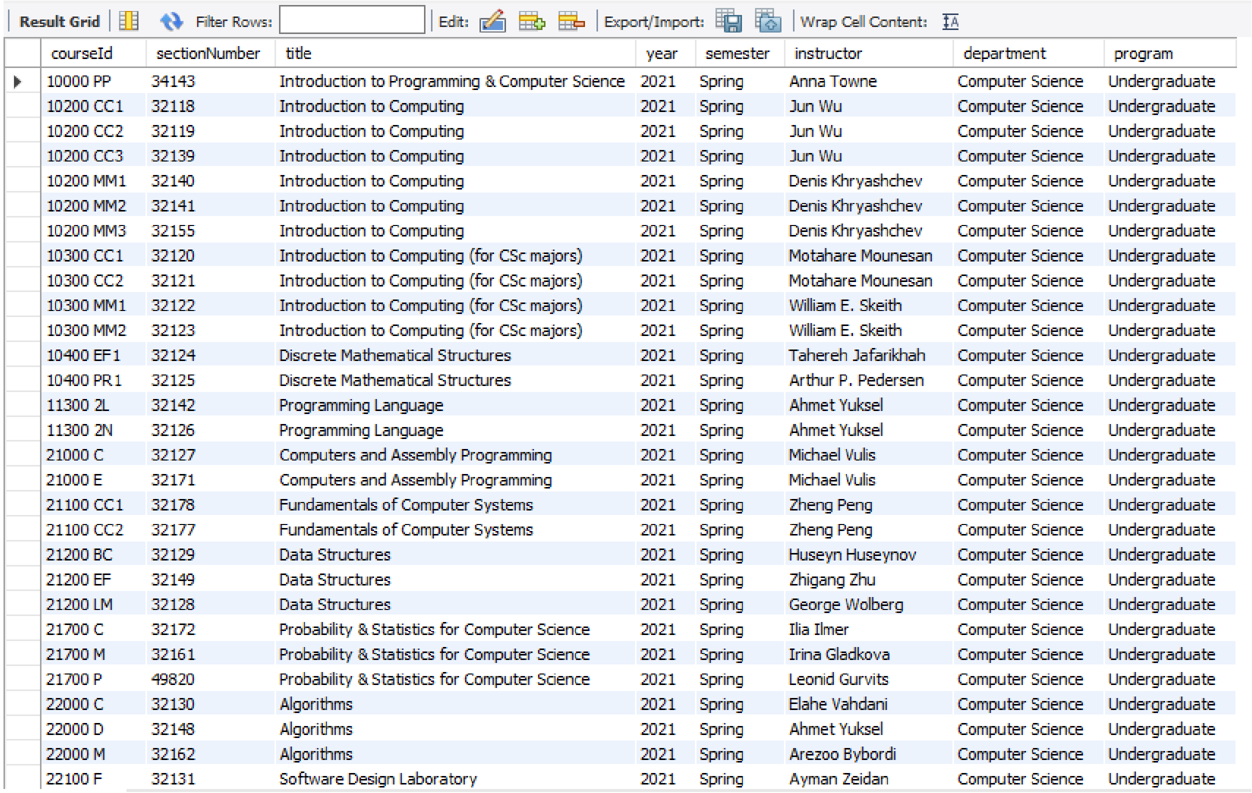
Aggregate Grades table creation in MySQL



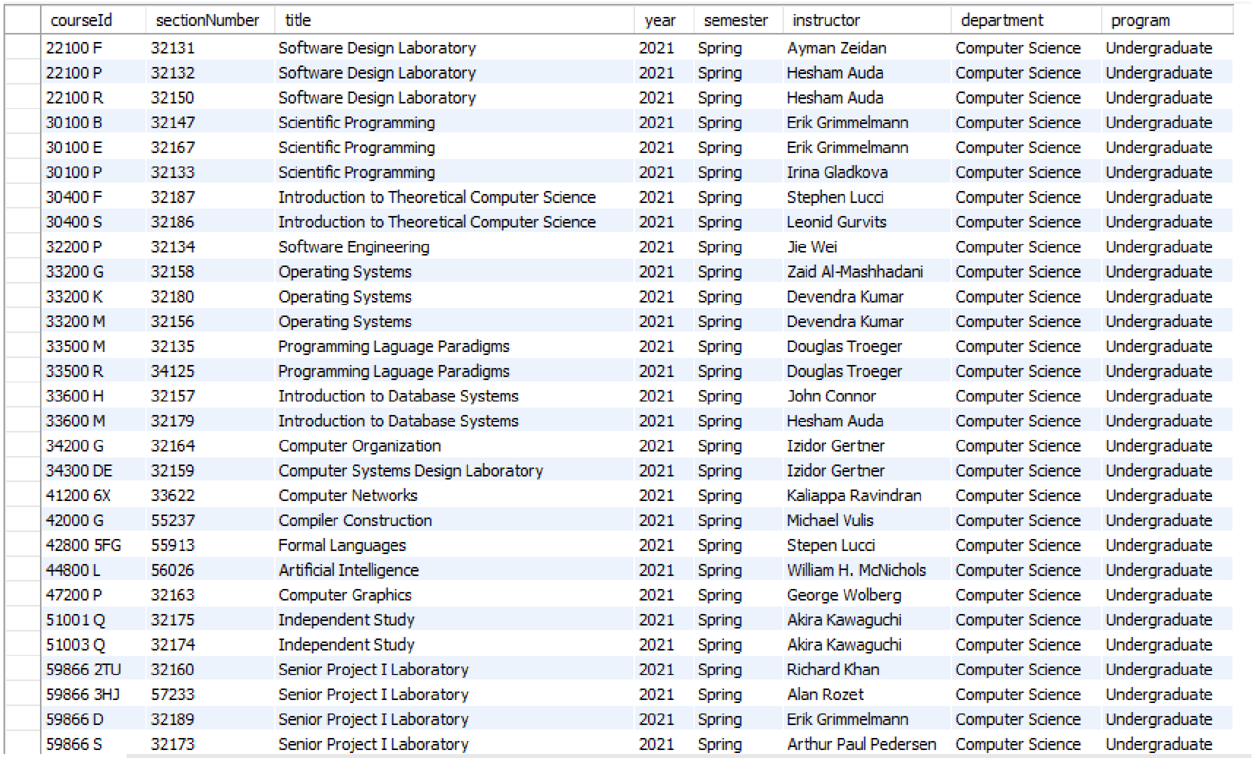
```
CREATE TABLE aggregategrades AS(
SELECT grade, COUNT(*)
FROM classes
WHERE courseId #gets grades from all classes and students
GROUP BY grade
ORDER BY COUNT(*) DESC); #primary key is set to 'grade' by default
```

All DDL and SQL statements used in MySQL are shown in the screenshots.

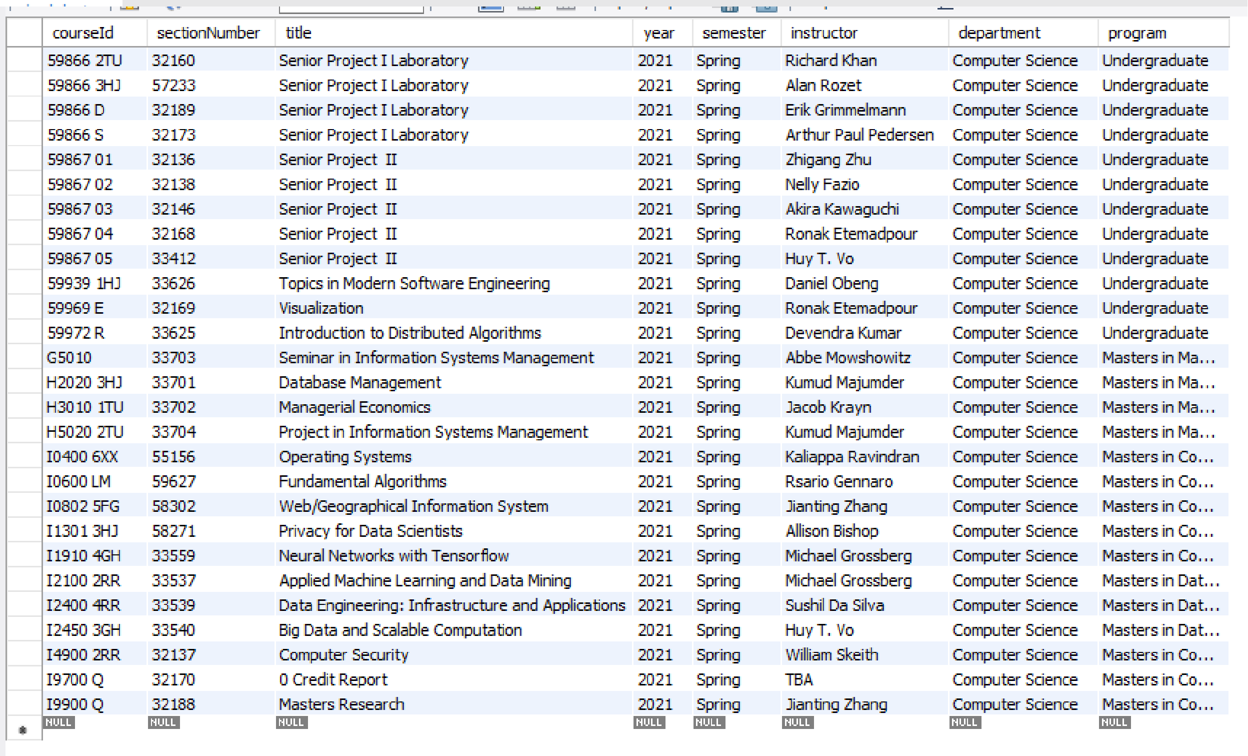
Part 4:



	courseId	sectionNumber	title	year	semester	instructor	department	program
▶	10000 PP	34143	Introduction to Programming & Computer Science	2021	Spring	Anna Towne	Computer Science	Undergraduate
	10200 CC1	32118	Introduction to Computing	2021	Spring	Jun Wu	Computer Science	Undergraduate
	10200 CC2	32119	Introduction to Computing	2021	Spring	Jun Wu	Computer Science	Undergraduate
	10200 CC3	32139	Introduction to Computing	2021	Spring	Jun Wu	Computer Science	Undergraduate
	10200 MM1	32140	Introduction to Computing	2021	Spring	Denis Khryashchev	Computer Science	Undergraduate
	10200 MM2	32141	Introduction to Computing	2021	Spring	Denis Khryashchev	Computer Science	Undergraduate
	10200 MM3	32155	Introduction to Computing	2021	Spring	Denis Khryashchev	Computer Science	Undergraduate
	10300 CC1	32120	Introduction to Computing (for CSc majors)	2021	Spring	Motahare Mounesan	Computer Science	Undergraduate
	10300 CC2	32121	Introduction to Computing (for CSc majors)	2021	Spring	Motahare Mounesan	Computer Science	Undergraduate
	10300 MM1	32122	Introduction to Computing (for CSc majors)	2021	Spring	William E. Skeith	Computer Science	Undergraduate
	10300 MM2	32123	Introduction to Computing (for CSc majors)	2021	Spring	William E. Skeith	Computer Science	Undergraduate
	10400 EF1	32124	Discrete Mathematical Structures	2021	Spring	Tahereh Jafarikhah	Computer Science	Undergraduate
	10400 PR1	32125	Discrete Mathematical Structures	2021	Spring	Arthur P. Pedersen	Computer Science	Undergraduate
	11300 2L	32142	Programming Language	2021	Spring	Ahmet Yuksel	Computer Science	Undergraduate
	11300 2N	32126	Programming Language	2021	Spring	Ahmet Yuksel	Computer Science	Undergraduate
	21000 C	32127	Computers and Assembly Programming	2021	Spring	Michael Vulis	Computer Science	Undergraduate
	21000 E	32171	Computers and Assembly Programming	2021	Spring	Michael Vulis	Computer Science	Undergraduate
	21100 CC1	32178	Fundamentals of Computer Systems	2021	Spring	Zheng Peng	Computer Science	Undergraduate
	21100 CC2	32177	Fundamentals of Computer Systems	2021	Spring	Zheng Peng	Computer Science	Undergraduate
	21200 BC	32129	Data Structures	2021	Spring	Huseyn Huseynov	Computer Science	Undergraduate
	21200 EF	32149	Data Structures	2021	Spring	Zhigang Zhu	Computer Science	Undergraduate
	21200 LM	32128	Data Structures	2021	Spring	George Wolberg	Computer Science	Undergraduate
	21700 C	32172	Probability & Statistics for Computer Science	2021	Spring	Ilia Ilmer	Computer Science	Undergraduate
	21700 M	32161	Probability & Statistics for Computer Science	2021	Spring	Irina Gladkova	Computer Science	Undergraduate
	21700 P	49820	Probability & Statistics for Computer Science	2021	Spring	Leonid Gurvits	Computer Science	Undergraduate
	22000 C	32130	Algorithms	2021	Spring	Elahe Vahdani	Computer Science	Undergraduate
	22000 D	32148	Algorithms	2021	Spring	Ahmet Yuksel	Computer Science	Undergraduate
	22000 M	32162	Algorithms	2021	Spring	Arezoo Bybordi	Computer Science	Undergraduate
	22100 F	32131	Software Design Laboratory	2021	Spring	Ayman Zeidan	Computer Science	Undergraduate

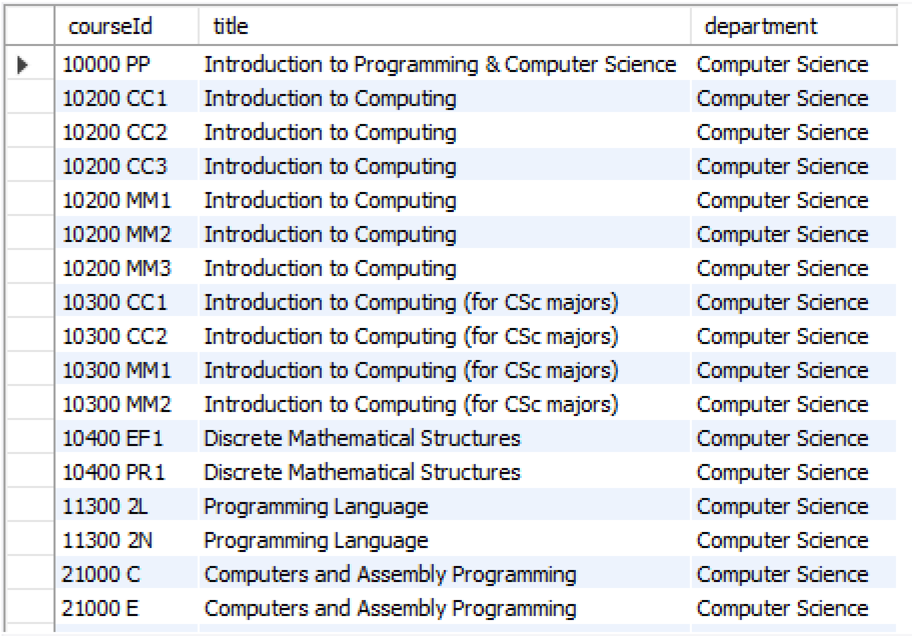


courseId	sectionNumber	title	year	semester	instructor	department	program
22100 F	32131	Software Design Laboratory	2021	Spring	Ayman Zeidan	Computer Science	Undergraduate
22100 P	32132	Software Design Laboratory	2021	Spring	Hesham Auda	Computer Science	Undergraduate
22100 R	32150	Software Design Laboratory	2021	Spring	Hesham Auda	Computer Science	Undergraduate
30100 B	32147	Scientific Programming	2021	Spring	Erik Grimmelmann	Computer Science	Undergraduate
30100 E	32167	Scientific Programming	2021	Spring	Erik Grimmelmann	Computer Science	Undergraduate
30100 P	32133	Scientific Programming	2021	Spring	Irina Gladkova	Computer Science	Undergraduate
30400 F	32187	Introduction to Theoretical Computer Science	2021	Spring	Stephen Lucci	Computer Science	Undergraduate
30400 S	32186	Introduction to Theoretical Computer Science	2021	Spring	Leonid Gurvits	Computer Science	Undergraduate
32200 P	32134	Software Engineering	2021	Spring	Jie Wei	Computer Science	Undergraduate
33200 G	32158	Operating Systems	2021	Spring	Zaid Al-Mashhadani	Computer Science	Undergraduate
33200 K	32180	Operating Systems	2021	Spring	Devendra Kumar	Computer Science	Undergraduate
33200 M	32156	Operating Systems	2021	Spring	Devendra Kumar	Computer Science	Undergraduate
33500 M	32135	Programming Laguage Paradigms	2021	Spring	Douglas Troeger	Computer Science	Undergraduate
33500 R	34125	Programming Laguage Paradigms	2021	Spring	Douglas Troeger	Computer Science	Undergraduate
33600 H	32157	Introduction to Database Systems	2021	Spring	John Connor	Computer Science	Undergraduate
33600 M	32179	Introduction to Database Systems	2021	Spring	Hesham Auda	Computer Science	Undergraduate
34200 G	32164	Computer Organization	2021	Spring	Izidor Gertner	Computer Science	Undergraduate
34300 DE	32159	Computer Systems Design Laboratory	2021	Spring	Izidor Gertner	Computer Science	Undergraduate
41200 6X	33622	Computer Networks	2021	Spring	Kaliappa Ravindran	Computer Science	Undergraduate
42000 G	55237	Compiler Construction	2021	Spring	Michael Vulis	Computer Science	Undergraduate
42800 5FG	55913	Formal Languages	2021	Spring	Stepen Lucci	Computer Science	Undergraduate
44800 L	56026	Artificial Intelligence	2021	Spring	William H. McNichols	Computer Science	Undergraduate
47200 P	32163	Computer Graphics	2021	Spring	George Wolberg	Computer Science	Undergraduate
51001 Q	32175	Independent Study	2021	Spring	Akira Kawaguchi	Computer Science	Undergraduate
51003 Q	32174	Independent Study	2021	Spring	Akira Kawaguchi	Computer Science	Undergraduate
59866 2TU	32160	Senior Project I Laboratory	2021	Spring	Richard Khan	Computer Science	Undergraduate
59866 3HJ	57233	Senior Project I Laboratory	2021	Spring	Alan Rozet	Computer Science	Undergraduate
59866 D	32189	Senior Project I Laboratory	2021	Spring	Erik Grimmelmann	Computer Science	Undergraduate
59866 S	32173	Senior Project I Laboratory	2021	Spring	Arthur Paul Pedersen	Computer Science	Undergraduate

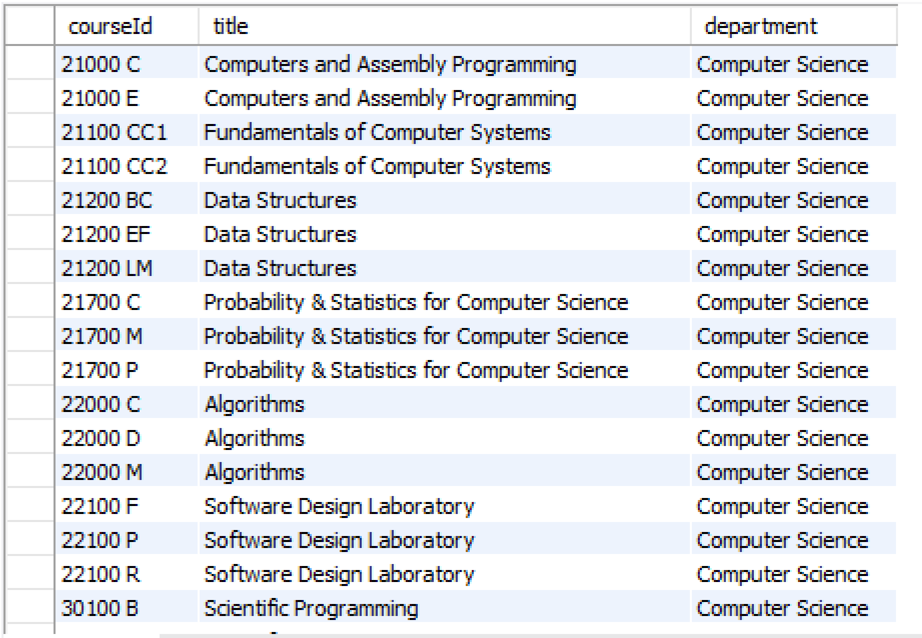


courseId	sectionNumber	title	year	semester	instructor	department	program
59866 2TU	32160	Senior Project I Laboratory	2021	Spring	Richard Khan	Computer Science	Undergraduate
59866 3HJ	57233	Senior Project I Laboratory	2021	Spring	Alan Rozet	Computer Science	Undergraduate
59866 D	32189	Senior Project I Laboratory	2021	Spring	Erik Grimmelmann	Computer Science	Undergraduate
59866 S	32173	Senior Project I Laboratory	2021	Spring	Arthur Paul Pedersen	Computer Science	Undergraduate
59867 01	32136	Senior Project II	2021	Spring	Zhigang Zhu	Computer Science	Undergraduate
59867 02	32138	Senior Project II	2021	Spring	Nelly Fazio	Computer Science	Undergraduate
59867 03	32146	Senior Project II	2021	Spring	Akira Kawaguchi	Computer Science	Undergraduate
59867 04	32168	Senior Project II	2021	Spring	Ronak Etemadpour	Computer Science	Undergraduate
59867 05	33412	Senior Project II	2021	Spring	Huy T. Vo	Computer Science	Undergraduate
59939 1HJ	33626	Topics in Modern Software Engineering	2021	Spring	Daniel Obeng	Computer Science	Undergraduate
59969 E	32169	Visualization	2021	Spring	Ronak Etemadpour	Computer Science	Undergraduate
59972 R	33625	Introduction to Distributed Algorithms	2021	Spring	Devendra Kumar	Computer Science	Undergraduate
G5010	33703	Seminar in Information Systems Management	2021	Spring	Abbe Mowshowitz	Computer Science	Masters in Ma...
H2020 3HJ	33701	Database Management	2021	Spring	Kumud Majumder	Computer Science	Masters in Ma...
H3010 1TU	33702	Managerial Economics	2021	Spring	Jacob Krayn	Computer Science	Masters in Ma...
H5020 2TU	33704	Project in Information Systems Management	2021	Spring	Kumud Majumder	Computer Science	Masters in Ma...
I0400 6XX	55156	Operating Systems	2021	Spring	Kaliappa Ravindran	Computer Science	Masters in Co...
I0600 LM	59627	Fundamental Algorithms	2021	Spring	Rsario Gennaro	Computer Science	Masters in Co...
I0802 5FG	58302	Web/Geographical Information System	2021	Spring	Jianting Zhang	Computer Science	Masters in Co...
I1301 3HJ	58271	Privacy for Data Scientists	2021	Spring	Allison Bishop	Computer Science	Masters in Co...
I1910 4GH	33559	Neural Networks with Tensorflow	2021	Spring	Michael Grossberg	Computer Science	Masters in Co...
I2100 2RR	33537	Applied Machine Learning and Data Mining	2021	Spring	Michael Grossberg	Computer Science	Masters in Dat...
I2400 4RR	33539	Data Engineering: Infrastructure and Applications	2021	Spring	Sushil Da Silva	Computer Science	Masters in Dat...
I2450 3GH	33540	Big Data and Scalable Computation	2021	Spring	Huy T. Vo	Computer Science	Masters in Dat...
I4900 2RR	32137	Computer Security	2021	Spring	William Skeith	Computer Science	Masters in Co...
I9700 Q	32170	0 Credit Report	2021	Spring	TBA	Computer Science	Masters in Co...
I9900 Q	32188	Masters Research	2021	Spring	Jianting Zhang	Computer Science	Masters in Co...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

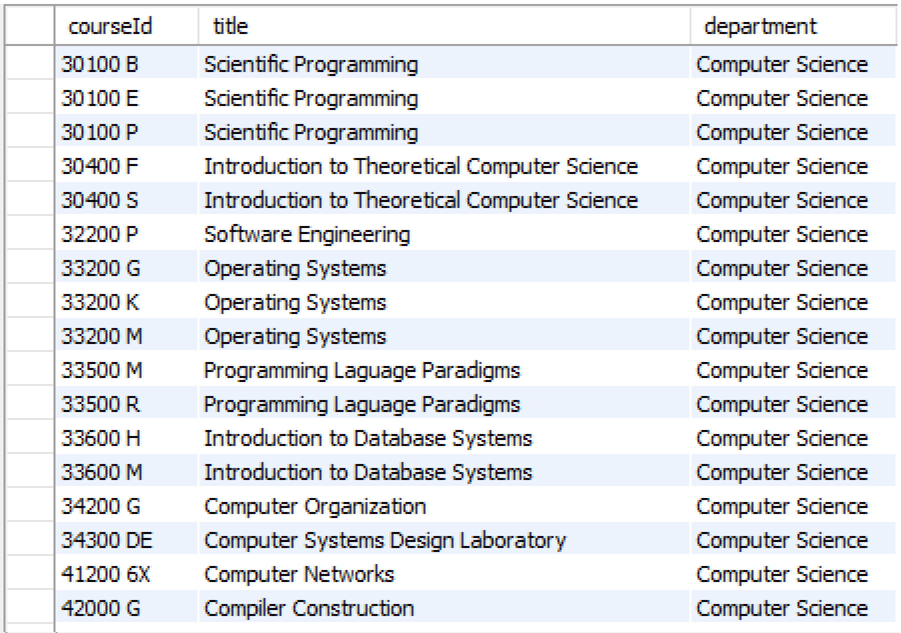
Create & Populate Schedule table using the data provided in the text file (I firstly made it into a .csv file using Excel and then transferred it over to MySQL).



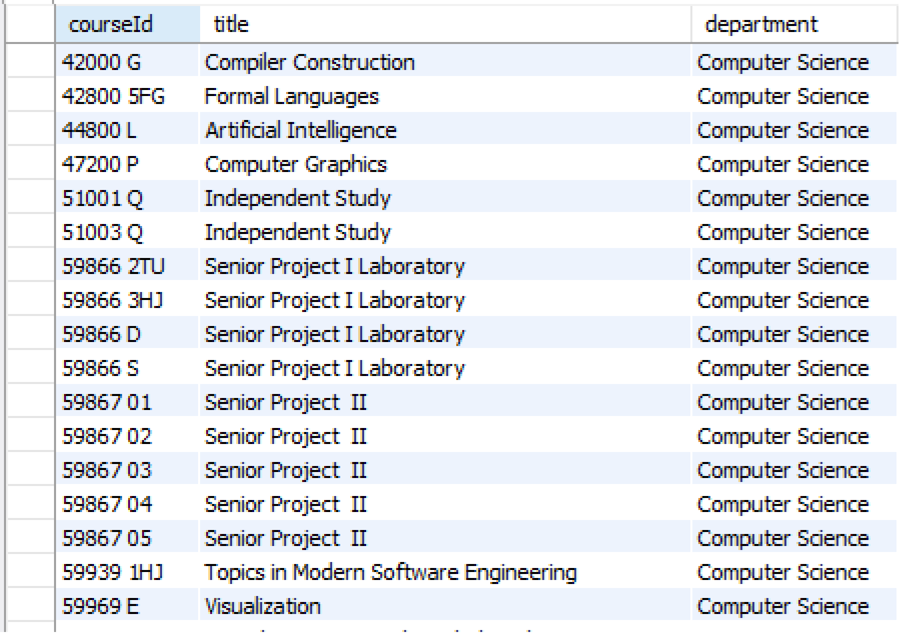
courseId	title	department
10000 PP	Introduction to Programming & Computer Science	Computer Science
10200 CC1	Introduction to Computing	Computer Science
10200 CC2	Introduction to Computing	Computer Science
10200 CC3	Introduction to Computing	Computer Science
10200 MM1	Introduction to Computing	Computer Science
10200 MM2	Introduction to Computing	Computer Science
10200 MM3	Introduction to Computing	Computer Science
10300 CC1	Introduction to Computing (for CSc majors)	Computer Science
10300 CC2	Introduction to Computing (for CSc majors)	Computer Science
10300 MM1	Introduction to Computing (for CSc majors)	Computer Science
10300 MM2	Introduction to Computing (for CSc majors)	Computer Science
10400 EF1	Discrete Mathematical Structures	Computer Science
10400 PR1	Discrete Mathematical Structures	Computer Science
11300 2L	Programming Language	Computer Science
11300 2N	Programming Language	Computer Science
21000 C	Computers and Assembly Programming	Computer Science
21000 E	Computers and Assembly Programming	Computer Science



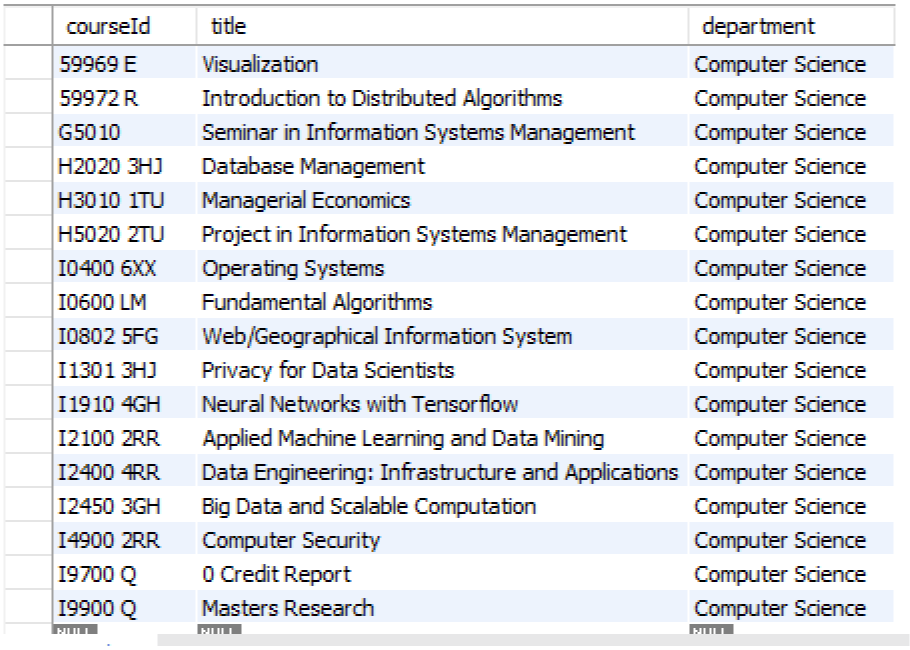
courseId	title	department
21000 C	Computers and Assembly Programming	Computer Science
21000 E	Computers and Assembly Programming	Computer Science
21100 CC1	Fundamentals of Computer Systems	Computer Science
21100 CC2	Fundamentals of Computer Systems	Computer Science
21200 BC	Data Structures	Computer Science
21200 EF	Data Structures	Computer Science
21200 LM	Data Structures	Computer Science
21700 C	Probability & Statistics for Computer Science	Computer Science
21700 M	Probability & Statistics for Computer Science	Computer Science
21700 P	Probability & Statistics for Computer Science	Computer Science
22000 C	Algorithms	Computer Science
22000 D	Algorithms	Computer Science
22000 M	Algorithms	Computer Science
22100 F	Software Design Laboratory	Computer Science
22100 P	Software Design Laboratory	Computer Science
22100 R	Software Design Laboratory	Computer Science
30100 B	Scientific Programming	Computer Science



courseId	title	department
30100 B	Scientific Programming	Computer Science
30100 E	Scientific Programming	Computer Science
30100 P	Scientific Programming	Computer Science
30400 F	Introduction to Theoretical Computer Science	Computer Science
30400 S	Introduction to Theoretical Computer Science	Computer Science
32200 P	Software Engineering	Computer Science
33200 G	Operating Systems	Computer Science
33200 K	Operating Systems	Computer Science
33200 M	Operating Systems	Computer Science
33500 M	Programming Laguage Paradigms	Computer Science
33500 R	Programming Laguage Paradigms	Computer Science
33600 H	Introduction to Database Systems	Computer Science
33600 M	Introduction to Database Systems	Computer Science
34200 G	Computer Organization	Computer Science
34300 DE	Computer Systems Design Laboratory	Computer Science
41200 6X	Computer Networks	Computer Science
42000 G	Compiler Construction	Computer Science

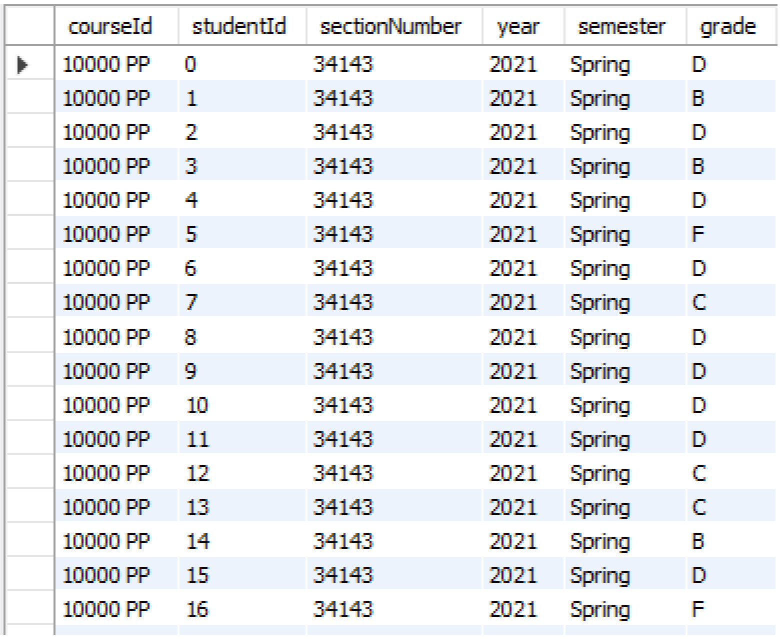


courseId	title	department
42000 G	Compiler Construction	Computer Science
42800 5FG	Formal Languages	Computer Science
44800 L	Artificial Intelligence	Computer Science
47200 P	Computer Graphics	Computer Science
51001 Q	Independent Study	Computer Science
51003 Q	Independent Study	Computer Science
59866 2TU	Senior Project I Laboratory	Computer Science
59866 3HJ	Senior Project I Laboratory	Computer Science
59866 D	Senior Project I Laboratory	Computer Science
59866 S	Senior Project I Laboratory	Computer Science
59867 01	Senior Project II	Computer Science
59867 02	Senior Project II	Computer Science
59867 03	Senior Project II	Computer Science
59867 04	Senior Project II	Computer Science
59867 05	Senior Project II	Computer Science
59939 1HJ	Topics in Modern Software Engineering	Computer Science
59969 E	Visualization	Computer Science

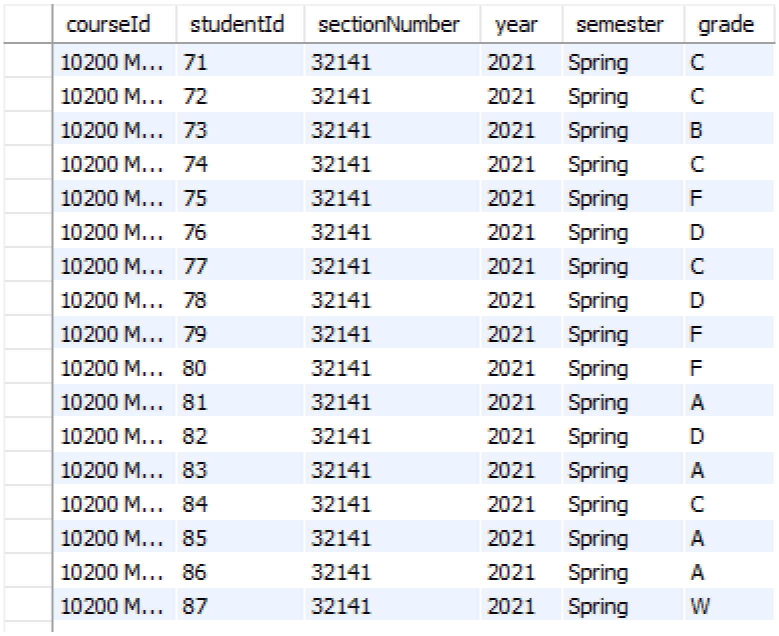


courseId	title	department
59969 E	Visualization	Computer Science
59972 R	Introduction to Distributed Algorithms	Computer Science
G5010	Seminar in Information Systems Management	Computer Science
H2020 3HJ	Database Management	Computer Science
H3010 1TU	Managerial Economics	Computer Science
H5020 2TU	Project in Information Systems Management	Computer Science
I0400 6XX	Operating Systems	Computer Science
I0600 LM	Fundamental Algorithms	Computer Science
I0802 5FG	Web/Geographical Information System	Computer Science
I1301 3HJ	Privacy for Data Scientists	Computer Science
I1910 4GH	Neural Networks with Tensorflow	Computer Science
I2100 2RR	Applied Machine Learning and Data Mining	Computer Science
I2400 4RR	Data Engineering: Infrastructure and Applications	Computer Science
I2450 3GH	Big Data and Scalable Computation	Computer Science
I4900 2RR	Computer Security	Computer Science
I9700 Q	0 Credit Report	Computer Science
I9900 Q	Masters Research	Computer Science
NULL	NULL	NULL

Courses table created and populated ^

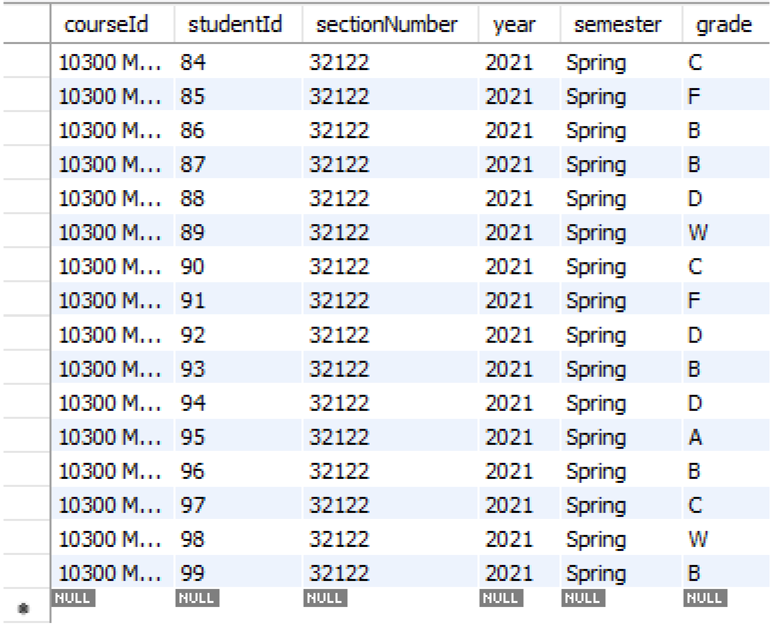


courseId	studentId	sectionNumber	year	semester	grade
10000 PP	0	34143	2021	Spring	D
10000 PP	1	34143	2021	Spring	B
10000 PP	2	34143	2021	Spring	D
10000 PP	3	34143	2021	Spring	B
10000 PP	4	34143	2021	Spring	D
10000 PP	5	34143	2021	Spring	F
10000 PP	6	34143	2021	Spring	D
10000 PP	7	34143	2021	Spring	C
10000 PP	8	34143	2021	Spring	D
10000 PP	9	34143	2021	Spring	D
10000 PP	10	34143	2021	Spring	D
10000 PP	11	34143	2021	Spring	D
10000 PP	12	34143	2021	Spring	C
10000 PP	13	34143	2021	Spring	C
10000 PP	14	34143	2021	Spring	B
10000 PP	15	34143	2021	Spring	D
10000 PP	16	34143	2021	Spring	F



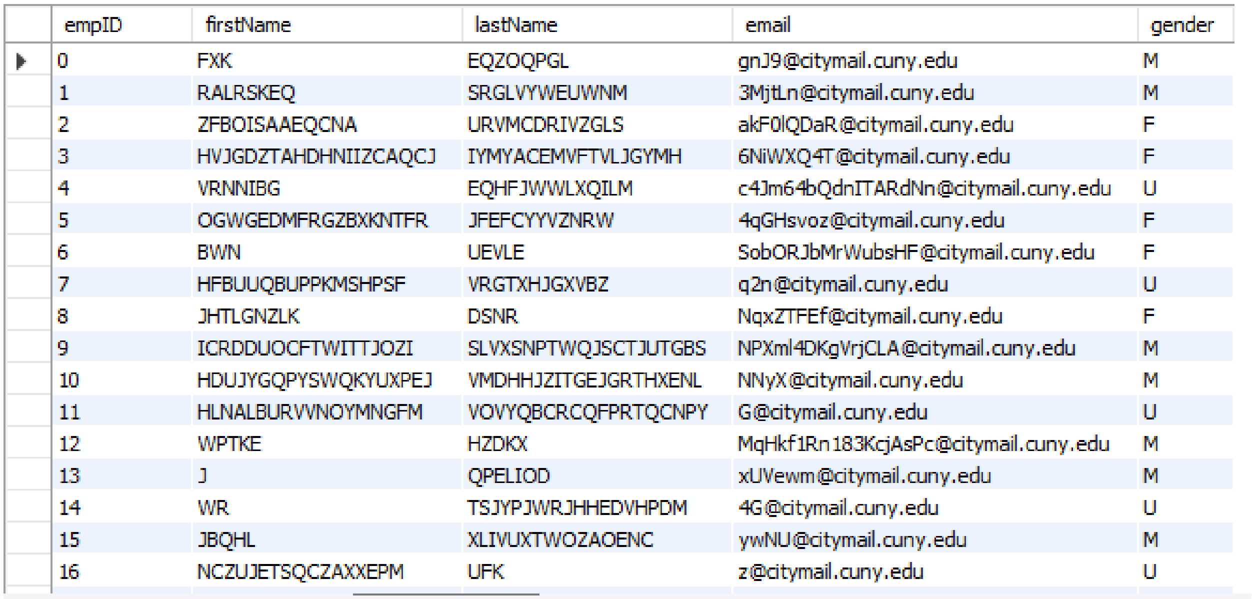
courseId	studentId	sectionNumber	year	semester	grade
10200 M...	71	32141	2021	Spring	C
10200 M...	72	32141	2021	Spring	C
10200 M...	73	32141	2021	Spring	B
10200 M...	74	32141	2021	Spring	C
10200 M...	75	32141	2021	Spring	F
10200 M...	76	32141	2021	Spring	D
10200 M...	77	32141	2021	Spring	C
10200 M...	78	32141	2021	Spring	D
10200 M...	79	32141	2021	Spring	F
10200 M...	80	32141	2021	Spring	F
10200 M...	81	32141	2021	Spring	A
10200 M...	82	32141	2021	Spring	D
10200 M...	83	32141	2021	Spring	A
10200 M...	84	32141	2021	Spring	C
10200 M...	85	32141	2021	Spring	A
10200 M...	86	32141	2021	Spring	A
10200 M...	87	32141	2021	Spring	W

(a couple of lines were skipped here for the sake of saved redundancy)

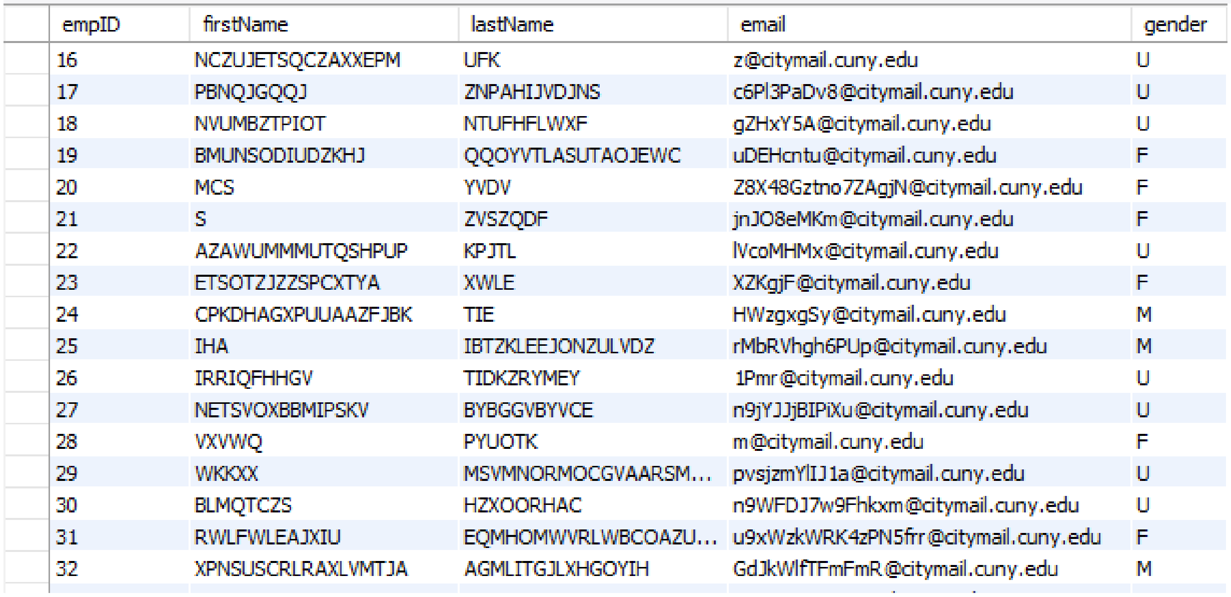


courseId	studentId	sectionNumber	year	semester	grade
10300 M...	84	32122	2021	Spring	C
10300 M...	85	32122	2021	Spring	F
10300 M...	86	32122	2021	Spring	B
10300 M...	87	32122	2021	Spring	B
10300 M...	88	32122	2021	Spring	D
10300 M...	89	32122	2021	Spring	W
10300 M...	90	32122	2021	Spring	C
10300 M...	91	32122	2021	Spring	F
10300 M...	92	32122	2021	Spring	D
10300 M...	93	32122	2021	Spring	B
10300 M...	94	32122	2021	Spring	D
10300 M...	95	32122	2021	Spring	A
10300 M...	96	32122	2021	Spring	B
10300 M...	97	32122	2021	Spring	C
10300 M...	98	32122	2021	Spring	W
10300 M...	99	32122	2021	Spring	B
NULL	NULL	NULL	NULL	NULL	NULL

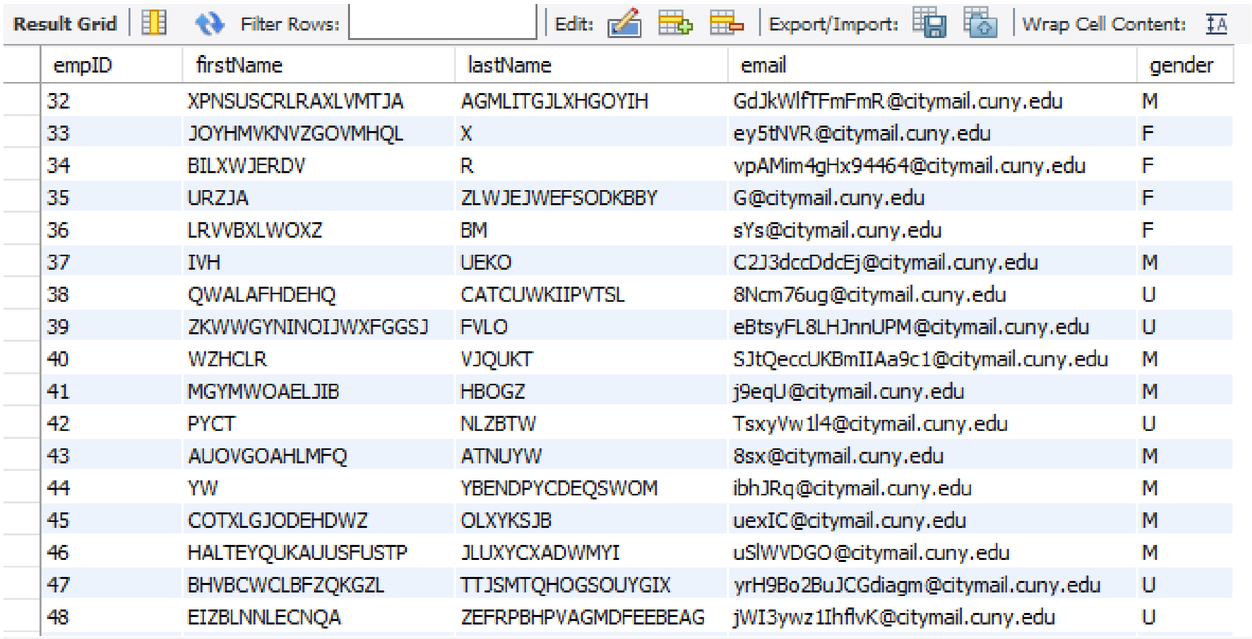
The Classes table created and populated ^ (not all is shown here, this table has 100 students for all classes and sections, hence the last studentId being 99 (id’s start from 0)).



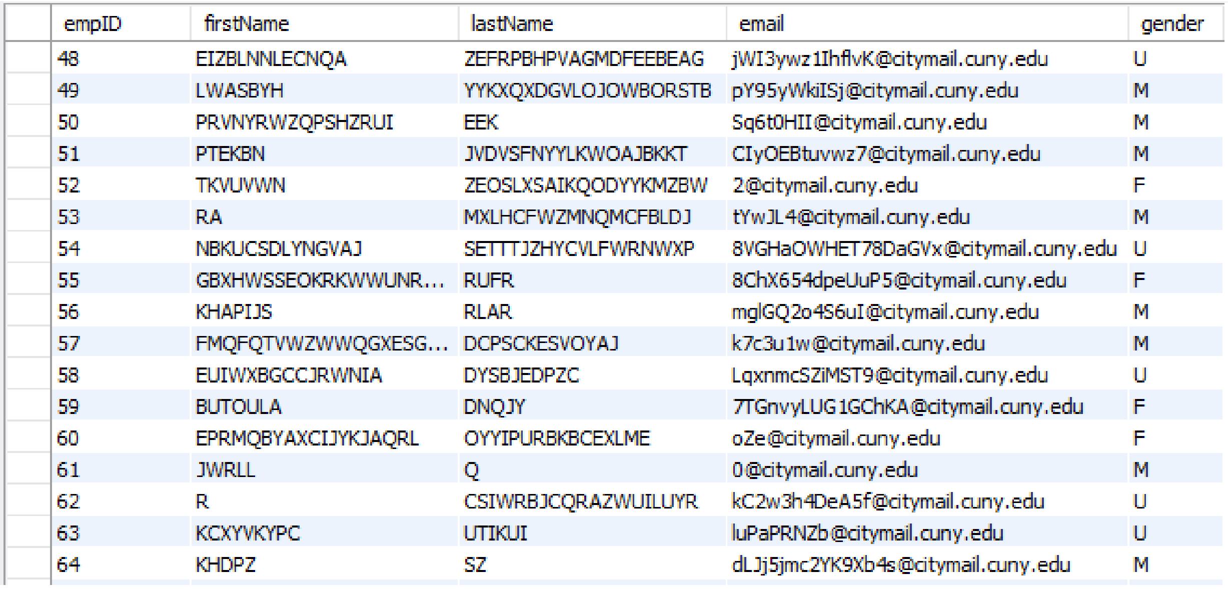
empID	firstName	lastName	email	gender
0	FXK	EQZOQPGL	gnJ9@citymail.cuny.edu	M
1	RALRSKEQ	SRGLVYWEUWNM	3MjtLn@citymail.cuny.edu	M
2	ZFBOISAAEQCNA	URVMCDRIVZGLS	akF0lQDaR@citymail.cuny.edu	F
3	HVJGDZTAHDHNIIZCAQCJ	IYMYACEMVFTVLJGYMH	6NiWXQ4T@citymail.cuny.edu	F
4	VRNNIBG	EQHFJWWLXQILM	c4Jm64bQdnITARdNn@citymail.cuny.edu	U
5	OGWGEDMFRGZBXKNTFR	JFEFCYYVZNRW	4qGHsvoz@citymail.cuny.edu	F
6	BWN	UEVLE	SobORJbMrWubsHF@citymail.cuny.edu	F
7	HFBUUQBUPPKMSHPSF	VRGTXHJGXVBZ	q2n@citymail.cuny.edu	U
8	JHTLGNZLK	DSNR	NqxZTFEf@citymail.cuny.edu	F
9	ICRDDUOCFTWITTJOZI	SLVXSNPTWQJSCTJUTGBS	NPXml4DKgVrjCLA@citymail.cuny.edu	M
10	HDUJYGQPYSWQKYUXPEJ	VMDHHJZITGEJGRTHXENL	NNyX@citymail.cuny.edu	M
11	HLNALBURVVNOYMNGFM	VOVYQBCRCQFPRTQCNPY	G@citymail.cuny.edu	U
12	WPTKE	HZDKX	MqHkf1Rn183KcjAsPc@citymail.cuny.edu	M
13	J	QPELIOD	xUVewm@citymail.cuny.edu	M
14	WR	TSJYPJWRJHHEDVHPDM	4G@citymail.cuny.edu	U
15	JBQHL	XLIVUXTWOZAOENC	ywNU@citymail.cuny.edu	M
16	NCZUJETSQCZAXXEPM	UFK	z@citymail.cuny.edu	U



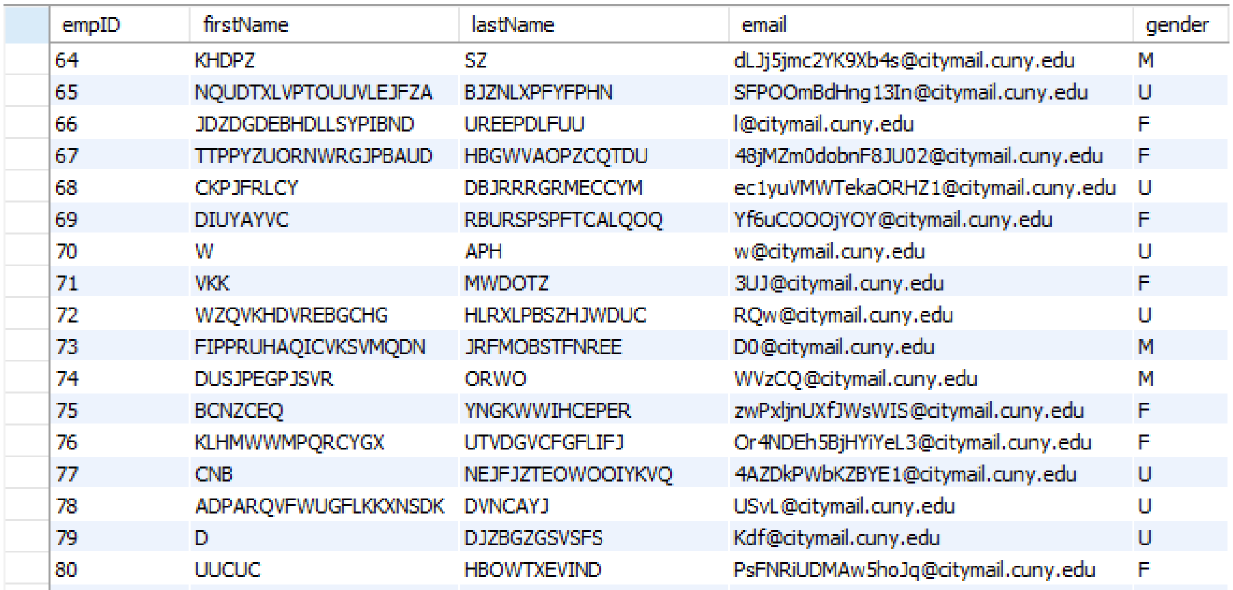
empID	firstName	lastName	email	gender
16	NCZUJETSQCZAXXEPM	UFK	z@citymail.cuny.edu	U
17	PBNQJGQQJ	ZNPAHIJVDJNS	c6Pl3PaDv8@citymail.cuny.edu	U
18	NVUMBZTPIOT	NTUFHFLWXF	gZHxY5A@citymail.cuny.edu	U
19	BMUNSODIUDZKHJ	QQOYVTLASUTAOJEWC	uDEHcntu@citymail.cuny.edu	F
20	MCS	YVDV	Z8X48Gztno7ZAgjN@citymail.cuny.edu	F
21	S	ZVSZQDF	jnJO8eMKm@citymail.cuny.edu	F
22	AZAWUMMMUTQSHPUP	KPJTL	lVcoMHMx@citymail.cuny.edu	U
23	ETSOTZJZZSPCXTYA	XWLE	XZKgjF@citymail.cuny.edu	F
24	CPKDHAGXPUUAAZFJBK	TIE	HWzgxgSy@citymail.cuny.edu	M
25	IHA	IBTZKLEEJONZULVDZ	rMbRVhgh6PUp@citymail.cuny.edu	M
26	IRRIQFHHGV	TIDKZRYMEY	1Pmr@citymail.cuny.edu	U
27	NETSVOXBBMIPSKV	BYBGGVBYVCE	n9jYJJjBIPiXu@citymail.cuny.edu	U
28	VXVWQ	PYUOTK	m@citymail.cuny.edu	F
29	WKKXX	MSVMNORMOCGVAARSM...	pvsjzmYlIJ1a@citymail.cuny.edu	U
30	BLMQTCZS	HZXOORHAC	n9WFDJ7w9Fhkxm@citymail.cuny.edu	U
31	RWLFWLEAJXIU	EQMHOMWVRLWBCOAZU...	u9xWzkWRK4zPN5frr@citymail.cuny.edu	F
32	XPNSUSCRLRAXLVMTJA	AGMLITGJLXHGOYIH	GdJkWlfTFmFmR@citymail.cuny.edu	M



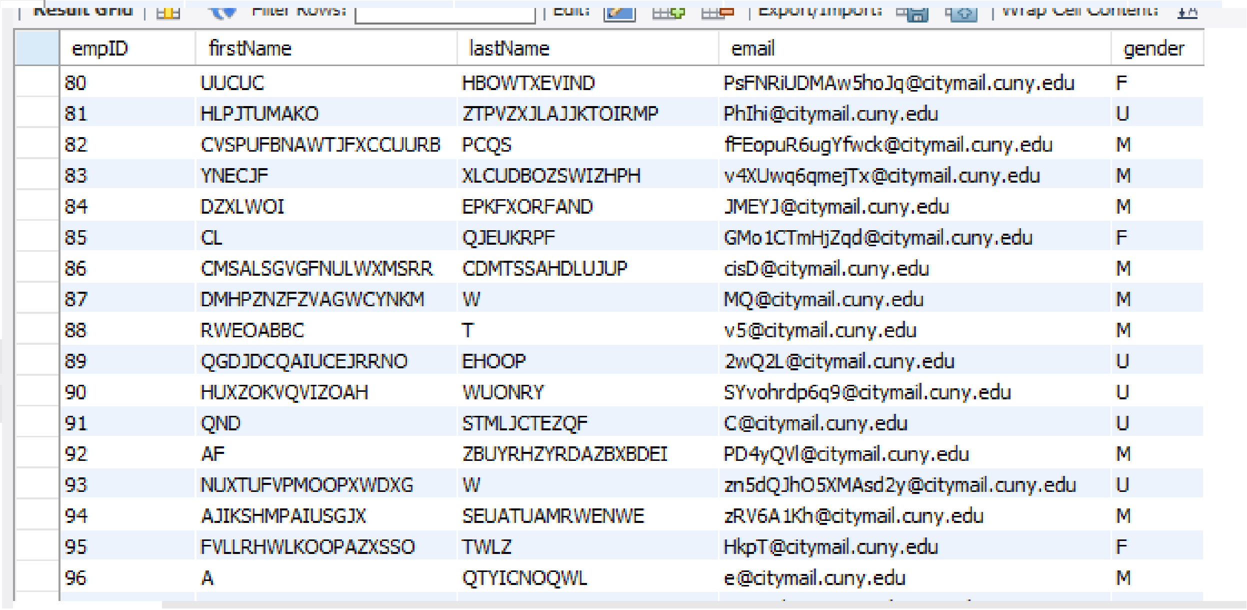
empID	firstName	lastName	email	gender
32	XPNSUSCRLRAXLVMTJA	AGMLITGJLXHGOYIH	GdJkWlfTFmFmR@citymail.cuny.edu	M
33	JOYHMVKNVZGOVMHQL	X	ey5tNVR@citymail.cuny.edu	F
34	BILXWJERDV	R	vpAMim4gHx94464@citymail.cuny.edu	F
35	URZJA	ZLWJEJWEFSODKBBY	G@citymail.cuny.edu	F
36	LRVVBXLWOXZ	BM	sYs@citymail.cuny.edu	F
37	IVH	UEKO	C2J3dccDdcEj@citymail.cuny.edu	M
38	QWALAFHDEHQ	CATCUWKIIPVTSL	8Ncm76ug@citymail.cuny.edu	U
39	ZKWWGYNINOIJWXFGGSJ	FVLO	eBtsyFL8LHJnnUPM@citymail.cuny.edu	U
40	WZHCLR	VJQUKT	SJtQeccUKBmIIAa9c1@citymail.cuny.edu	M
41	MGYMWOAELJIB	HBOGZ	j9eqU@citymail.cuny.edu	M
42	PYCT	NLZBTW	TsxyVw1l4@citymail.cuny.edu	U
43	AUOVGOAHLMFQ	ATNUYW	8sx@citymail.cuny.edu	M
44	YW	YBENDPYCDEQSWOM	ibhJRq@citymail.cuny.edu	M
45	COTXLGJODEHDWZ	OLXYKSJB	uexIC@citymail.cuny.edu	M
46	HALTEYQUKAUUSFUSTP	JLUXYCXADWMYI	uSlWVDGO@citymail.cuny.edu	M
47	BHVBCWCLBFZQKGZL	TTJSMTQHOGSOUYGIX	yrH9Bo2BuJCGdiagm@citymail.cuny.edu	U
48	EIZBLNNLECNQA	ZEFRPBHPVAGMDFEEBEAG	jWI3ywz1IhflvK@citymail.cuny.edu	U



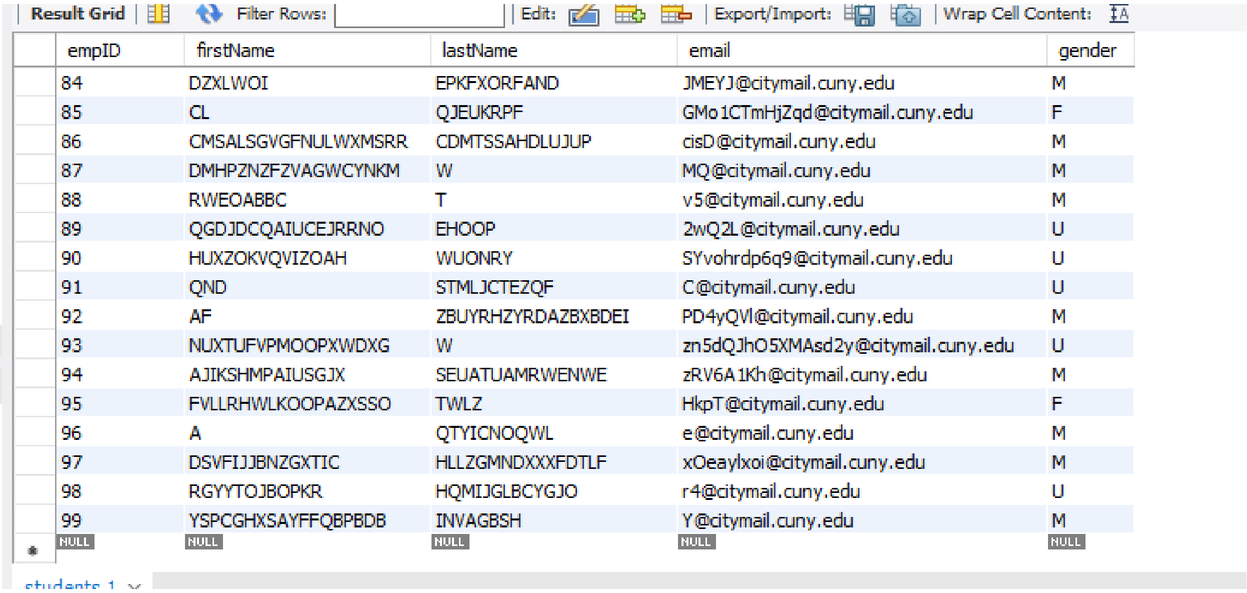
empID	firstName	lastName	email	gender
48	EIZBLNNLECNQA	ZEFRPBHPVAGMDFEEBEAG	jWI3ywz1IhflvK@citymail.cuny.edu	U
49	LWASBYH	YYKXQXDGVLOJOWBORSTB	pY95yWkiISj@citymail.cuny.edu	M
50	PRVNYRWZQPSHZRUI	EEK	Sq6t0HII@citymail.cuny.edu	M
51	PTEKBN	JVDVSFNYYLKWOAJBKKT	CIyOEBtuvwz7@citymail.cuny.edu	M
52	TKVUVWN	ZEOSLXSAIKQODYYKMZBW	2@citymail.cuny.edu	F
53	RA	MXLHCFWZMNQMCFBLDJ	tYwJL4@citymail.cuny.edu	M
54	NBKUCSDLYNGVAJ	SETTTJZHYCVLFWRNWXP	8VGHaOWHET78DaGVx@citymail.cuny.edu	U
55	GBXHWSSEOKRKWWUNR...	RUFR	8ChX654dpeUuP5@citymail.cuny.edu	F
56	KHAPIJS	RLAR	mglGQ2o4S6uI@citymail.cuny.edu	M
57	FMQFQTVWZWWQGXESG...	DCPSCKESVOYAJ	k7c3u1w@citymail.cuny.edu	M
58	EUIWXBGCCJRWNIA	DYSBJEDPZC	LqxnmcSZiMST9@citymail.cuny.edu	U
59	BUTOULA	DNQJY	7TGnvyLUG1GChKA@citymail.cuny.edu	F
60	EPRMQBYAXCIJYKJAQRL	OYYIPURBKBCEXLME	oZe@citymail.cuny.edu	F
61	JWRLL	Q	0@citymail.cuny.edu	M
62	R	CSIWRBJCQRAZWUILUYR	kC2w3h4DeA5f@citymail.cuny.edu	U
63	KCXYVKYPC	UTIKUI	luPaPRNZb@citymail.cuny.edu	U
64	KHDPZ	SZ	dLJj5jmc2YK9Xb4s@citymail.cuny.edu	M



empID	firstName	lastName	email	gender
64	KHDPZ	SZ	dLJj5jmc2YK9Xb4s@citymail.cuny.edu	M
65	NQUDTXLVPTOUUVLEJFZA	BJZNLXPFYFPHN	SFPOOmBdHng13In@citymail.cuny.edu	U
66	JDZDGDEBHDLLSYPIBND	UREEPDLFUU	l@citymail.cuny.edu	F
67	TTPPYZUORNWRGJPBAUD	HBGWVAOPZCQTDU	48jMZm0dobnF8JU02@citymail.cuny.edu	F
68	CKPJFRLCY	DBJRRRGRMECCYM	ec1yuVMWTekaORHZ1@citymail.cuny.edu	U
69	DIUYAYVC	RBURSPSPFTCALQOQ	Yf6uCOOOjYOY@citymail.cuny.edu	F
70	W	APH	w@citymail.cuny.edu	U
71	VKK	MWDOTZ	3UJ@citymail.cuny.edu	F
72	WZQVKHDVREBGCHG	HLRXLPBSZHJWDUC	RQw@citymail.cuny.edu	U
73	FIPPRUHAQICVKSVMQDN	JRFMOBSTFNREE	D0@citymail.cuny.edu	M
74	DUSJPEGPJSVR	ORWO	WVzCQ@citymail.cuny.edu	M
75	BCNZCEQ	YNGKWWIHCEPER	zwPxljnUXfJWsWIS@citymail.cuny.edu	F
76	KLHMWWMPQRCYGX	UTVDGVCFGFLIFJ	Or4NDEh5BjHYiYeL3@citymail.cuny.edu	F
77	CNB	NEJFJZTEOWOOIYKVQ	4AZDkPWbKZBYE1@citymail.cuny.edu	U
78	ADPARQVFWUGFLKKXNSDK	DVNCAYJ	USvL@citymail.cuny.edu	U
79	D	DJZBGZGSVSFS	Kdf@citymail.cuny.edu	U
80	UUCUC	HBOWTXEVIND	PsFNRiUDMAw5hoJq@citymail.cuny.edu	F

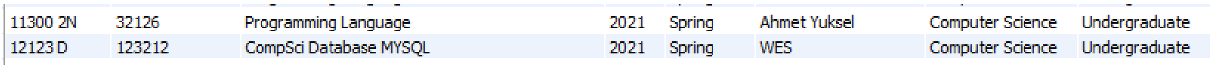


empID	firstName	lastName	email	gender
80	UUCUC	HBOWTXEVIND	PsFNRiUDMAw5hoJq@citymail.cuny.edu	F
81	HLPJTUMAKO	ZTPVZXJLAJJKTOIRMP	PhIhi@citymail.cuny.edu	U
82	CVSPUFBNAWTJFXCCUURB	PCQS	fFEopuR6ugYfwck@citymail.cuny.edu	M
83	YNECJF	XLCUDBOZSWIZHPH	v4XUwq6qmejTx@citymail.cuny.edu	M
84	DZXLWOI	EPKFXORFAND	JMEYJ@citymail.cuny.edu	M
85	CL	QJEUKRPF	GMo1CTmHjZqd@citymail.cuny.edu	F
86	CMSALSGVGFNULWXMSRR	CDMTSSAHDLUJUP	cisD@citymail.cuny.edu	M
87	DMHPZNZFZVAGWCYNKM	W	MQ@citymail.cuny.edu	M
88	RWEOABBC	T	v5@citymail.cuny.edu	M
89	QGDJDCQAIUCEJRRNO	EHOOP	2wQ2L@citymail.cuny.edu	U
90	HUXZOKVQVIZOAH	WUONRY	SYvohrdp6q9@citymail.cuny.edu	U
91	QND	STMLJCTEZQF	C@citymail.cuny.edu	U
92	AF	ZBUYRHZYRDAZBXBDEI	PD4yQVl@citymail.cuny.edu	M
93	NUXTUFVPMOOPXWDXG	W	zn5dQJhO5XMAsd2y@citymail.cuny.edu	U
94	AJIKSHMPAIUSGJX	SEUATUAMRWENWE	zRV6A1Kh@citymail.cuny.edu	M
95	FVLLRHWLKOOPAZXSSO	TWLZ	HkpT@citymail.cuny.edu	F
96	A	QTYICNOQWL	e@citymail.cuny.edu	M



empID	firstName	lastName	email	gender
84	DZXLWOI	EPKFXORFAND	JMEYJ@citymail.cuny.edu	M
85	CL	QJEUKRPF	GMo1CTmHjZqd@citymail.cuny.edu	F
86	CMSALSGVGFNULWXMSRR	CDMTSSAHDLUJUP	cisD@citymail.cuny.edu	M
87	DMHPZNZFZVAGWCYNKM	W	MQ@citymail.cuny.edu	M
88	RWEOABBC	T	v5@citymail.cuny.edu	M
89	QGDJDCQAIUCEJRRNO	EHOOP	2wQ2L@citymail.cuny.edu	U
90	HUXZOKVQVIZOAH	WUONRY	SYvohrdp6q9@citymail.cuny.edu	U
91	QND	STMLJCTEZQF	C@citymail.cuny.edu	U
92	AF	ZBUYRHZYRDAZBXBDEI	PD4yQVl@citymail.cuny.edu	M
93	NUXTUFVPMOOPXWDXG	W	zn5dQJhO5XMAsd2y@citymail.cuny.edu	U
94	AJIKSHMPAIUSGJX	SEUATUAMRWENWE	zRV6A1Kh@citymail.cuny.edu	M
95	FVLLRHWLKOOPAZXSSO	TWLZ	HkpT@citymail.cuny.edu	F
96	A	QTYICNOQWL	e@citymail.cuny.edu	M
97	DSVFIJJBNZGXTIC	HLLZGMNDXXXFDTLF	xOeaylxoi@citymail.cuny.edu	M
98	RGYYTOJBOPKR	HQMIJGLBCYGJO	r4@citymail.cuny.edu	U
99	YSPCGHXSAYFFQBPBDB	INVAGBSH	Y@citymail.cuny.edu	M
NULL	NULL	NULL	NULL	NULL

Students table created and populated with my own randomized data

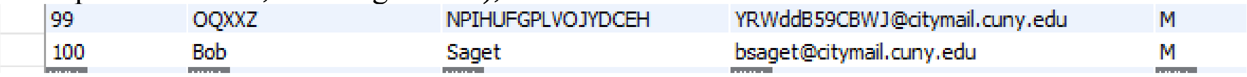


11300 2N	32126	Programming Language	2021	Spring	Ahmet Yuksel	Computer Science	Undergraduate
12123 D	123212	CompSci Database MYSQL	2021	Spring	WES	Computer Science	Undergraduate

Row inserted into "schedule" table by the use of this line:

```
Schedule.insertRow(con, "12123 D", "123212", "CompSci Database MYSQL", "2021", "Spring", "WES", "Computer Science", "Undergraduate");
```

Followed SQL formatting: INSERT INTO schedule (courseId, sectionNumber, title, year, semester, instructor, department, program)
VALUES ("12123 D", "123212", CompSci Database MYSQL", 2021, "Spring", "WES", "Computer Science", "Undergraduate);

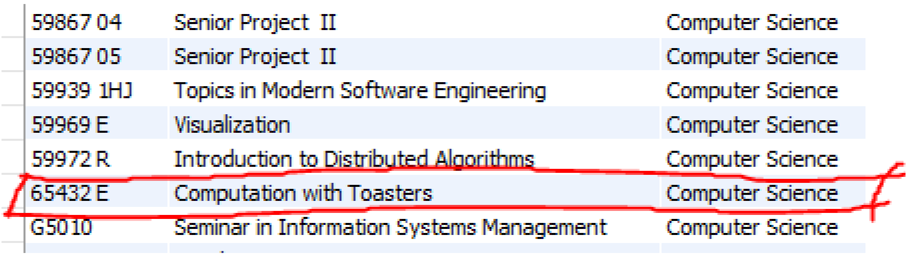


99	OQXXZ	NPIHUFGPLVOJYDCEH	YRWddB59CBWJ@citymail.cuny.edu	M
100	Bob	Saget	bsaget@citymail.cuny.edu	M

NOTE: new student table, still with 100 students, but the randomized function made a different table set. However, it follows the same pattern as the screenshots above.
This was inserting a new student into the table "students" using this line of code:

```
Students.insertRow(con, "100", "Bob", "Saget", "bsaget@citymail.cuny.edu", "M");
```

Followed SQL formatting: INSERT INTO schedule (empID, firstName, lastName, email, gender)
VALUES (100, "Bob", "Saget", "bsaget@citymail.cuny.edu", "M");

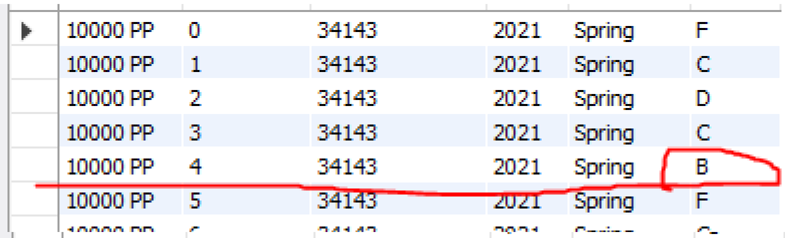


59867 04	Senior Project II	Computer Science
59867 05	Senior Project II	Computer Science
59939 1HJ	Topics in Modern Software Engineering	Computer Science
59969 E	Visualization	Computer Science
59972 R	Introduction to Distributed Algorithms	Computer Science
65432 E	Computation with Toasters	Computer Science
G5010	Seminar in Information Systems Management	Computer Science

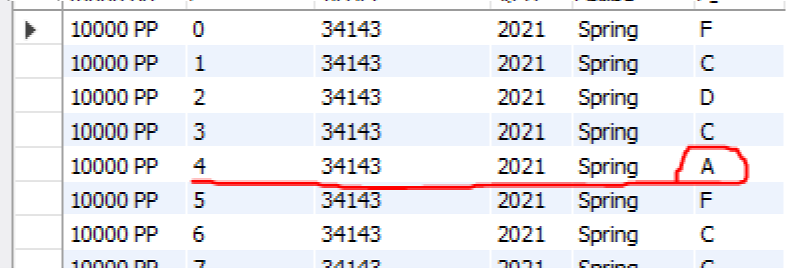
This was inserting a new row into the table "courses" using this line of code:

```
Courses.insertRow(con, "65432 E", "Computation with Toasters", "Computer Science");
```

Followed SQL formatting: INSERT INTO courses (courseId, title, department) VALUES (“65432 E”, “Computation with Toasters”, “Computer Science”);



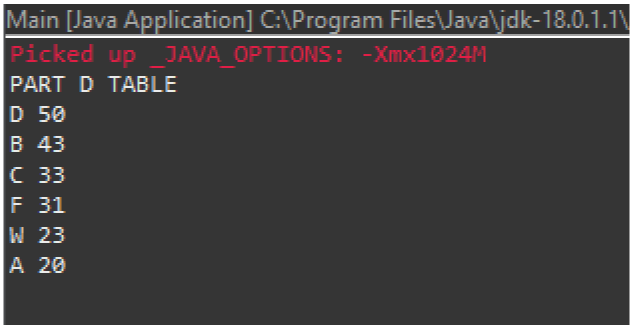
▶	10000 PP	0	34143	2021	Spring	F
	10000 PP	1	34143	2021	Spring	C
	10000 PP	2	34143	2021	Spring	D
	10000 PP	3	34143	2021	Spring	C
	10000 PP	4	34143	2021	Spring	B
	10000 PP	5	34143	2021	Spring	F



▶	10000 PP	0	34143	2021	Spring	F
	10000 PP	1	34143	2021	Spring	C
	10000 PP	2	34143	2021	Spring	D
	10000 PP	3	34143	2021	Spring	C
	10000 PP	4	34143	2021	Spring	A
	10000 PP	5	34143	2021	Spring	F
	10000 PP	6	34143	2021	Spring	C

Updating the grade in table classes by the use of this line: `Classes.setGrade(con, 'A', 4, "10000 PP");`

Followed SQL formatting: UPDATE classes SET grade = 'A' WHERE studentId = "4" AND courseId = "10000 PP";



```
Main [Java Application] C:\Program Files\Java\jdk-18.0.1.1\
Picked up _JAVA_OPTIONS: -Xmx1024M
PART D TABLE
D 50
B 43
C 33
F 31
W 23
A 20
```

Information printed out for Part D (using GROUP BY)

Used the following SQL formatting: SELECT grade, COUNT(*) FROM classes WHERE courseId in (“33600 H”, “33600 M”) GROUP BY grade ORDER BY COUNT(*) DESC;

Part E:

All of the above is created by the Java application and uses PreparedStatement objects for the execution of DDL statements and SQL queries.

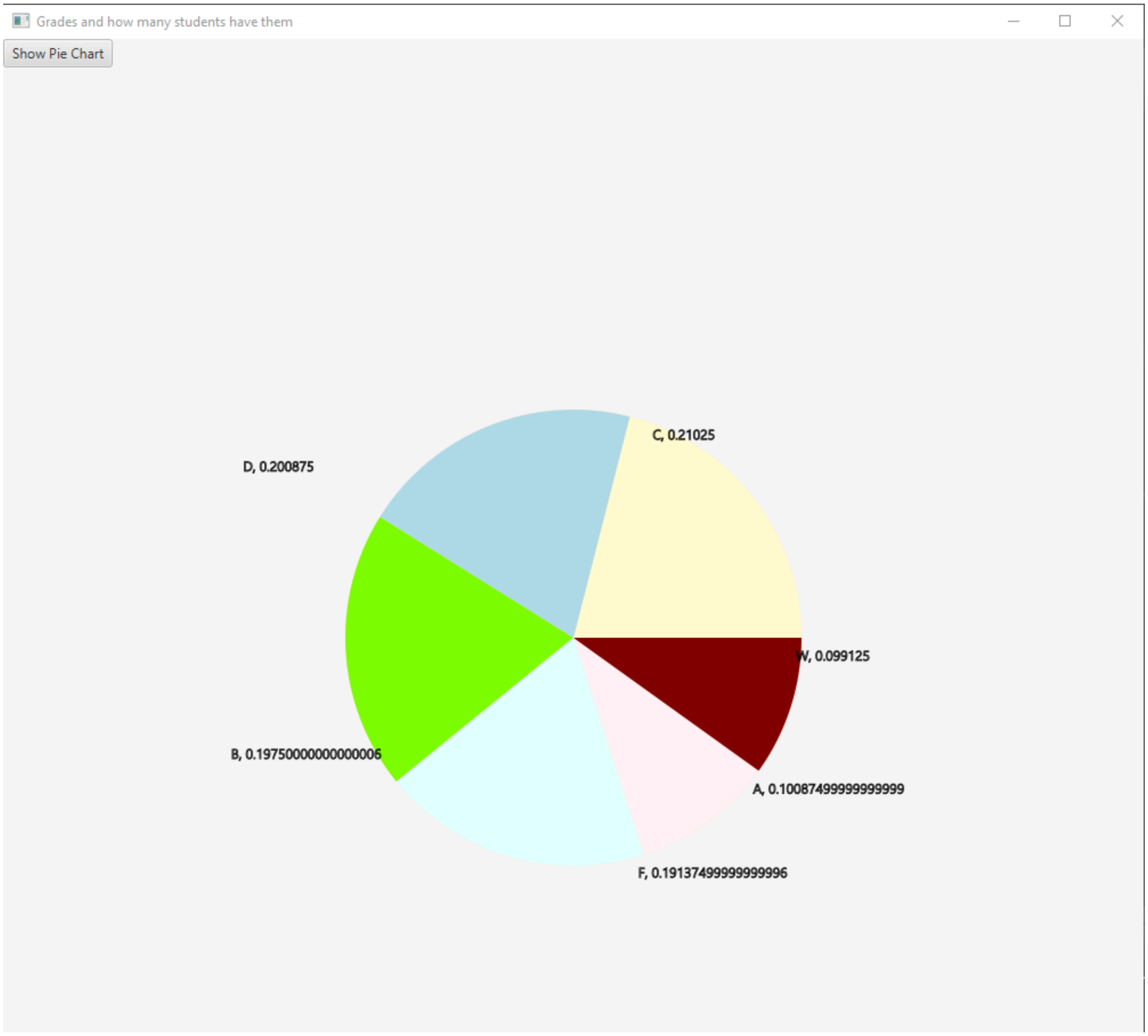
Part F:

The Database class itself is in the code and has the inner classes Schedule, Students, Courses, Classes, and AggregateGrades inside of it (which respond to the corresponding tables aswell). The database itself actually sets up the connection to the RDMBS as its constructor takes in strings that help set up the connection: the url, the username, and the password. The inner classes are used for creating, populating, and querying the database tables.

Part G:

Class Database implements interface ClassScheduleInterface, but the interface is empty in my case. I could not get to it on time.

Part H:

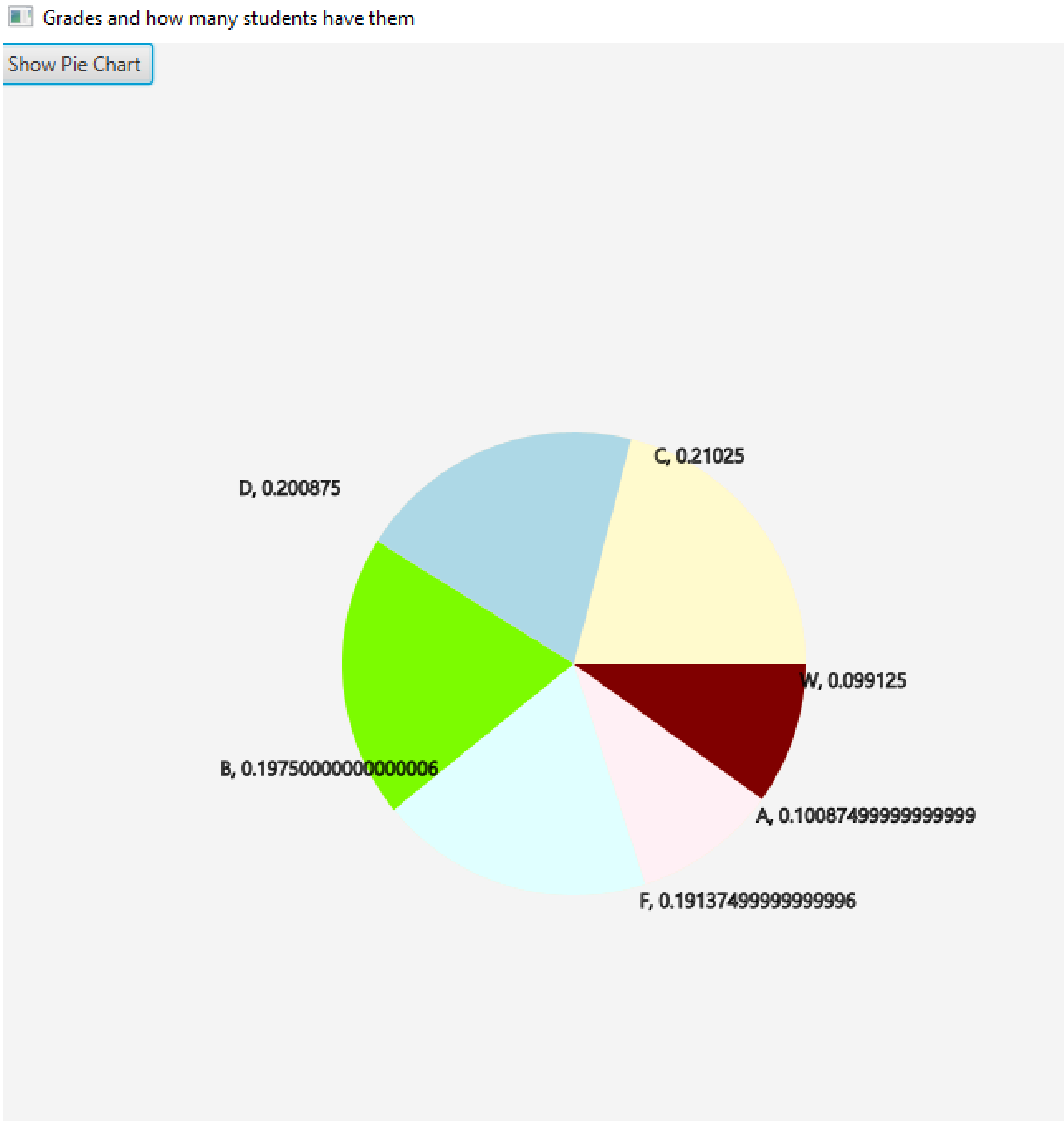


Shown is the resulting Pie Chart showing the portion of students for each letter grade. The pie chart has different colors for each segment, and legends, but does not display the number of students (but rather a frequency), and the grades are not displayed in alphabetic order.

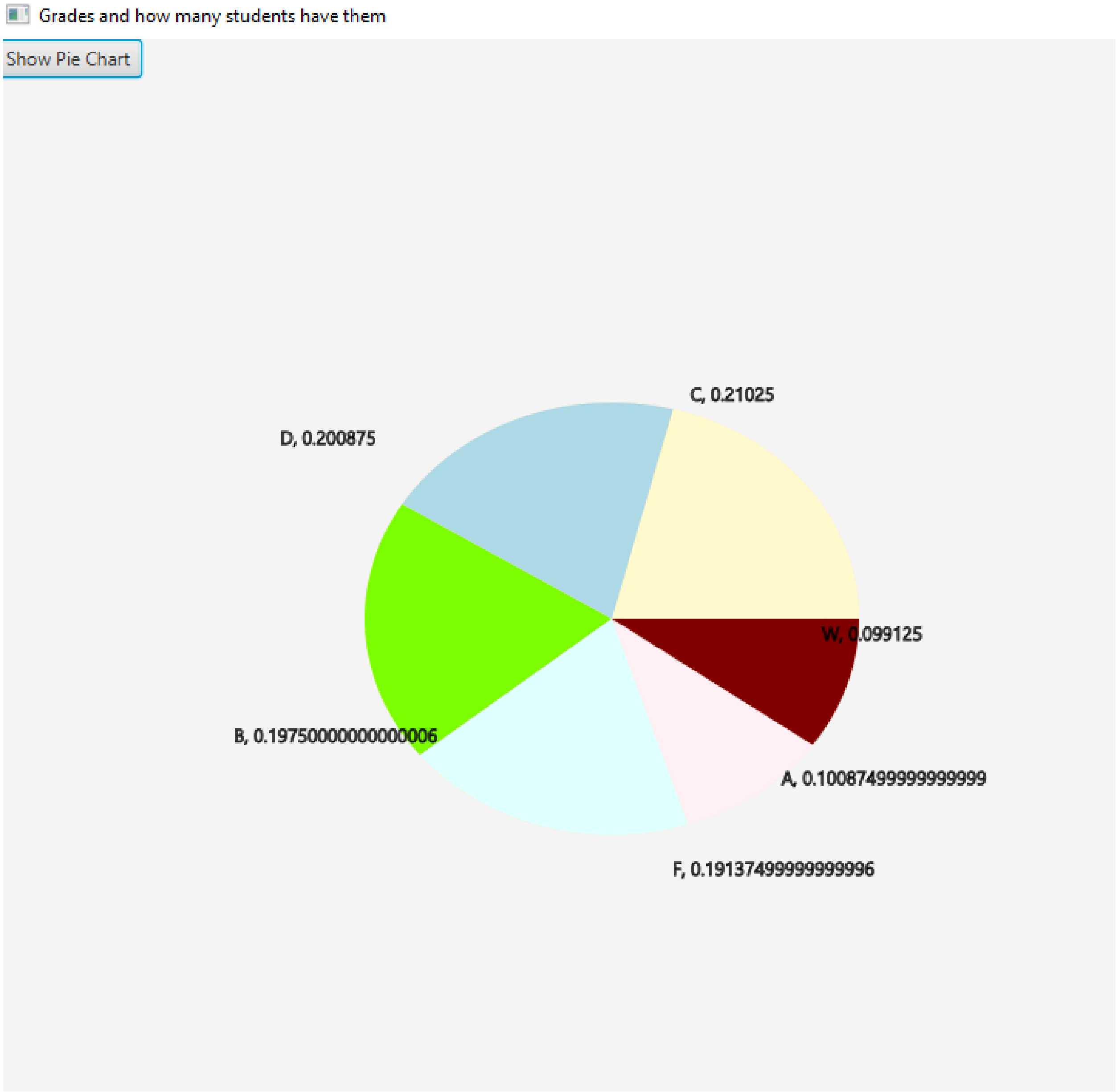
Part 3 of the PDF:
Sample input tables are shown, as well as output table for part D of the assignment. Corresponding pie chart was shown for 200 students (100 input into the code but turned into 200 because of how the input table was randomized for the classes table) in classes with section ID's "33600 H" and "33600 M" (essentially that of part D, but made within the Database class as an

AggregateGrade table – “Database students = new Database("schedule", "ScheduleSpring2021new.csv", "students", 100, "courses", "classes", "aggregategrades", "33600 H", "33600 M", "NULL", "NULL", "NULL", "jdbc:mysql://localhost:3306/***", "***", "***");”

Part 4 of the PDF:



Canvas size of 700x700



Canvas size of 800 x 700

Part 5 of the PDF:

The FileReader class was imported so that the .txt file of "grades" could be read into the program through the use of a FileReader instance, which took in the path to the .txt file, and allowed for the Java program to parse through the entire text that was created. It was also used to use the read method.

The FileWriter class was imported to allow the "grades" text file to be written into my computer through the use of a FileWriter instance in the gradesToFile void method of Classes.

The IOException class was imported because it was used in the HistogramAlphaBet constructor if the path to the .txt file was not valid, and also in the lambda expression in the main program for the button in case the program didn't work out because of the FilePath that was input into the HistogramAlphaBet constructor. The class was used simply to check for exceptions or failures in the I/O operations, specifically in the path checking operation of the FileReader.

The DriverManager class was imported from java.sql to allow for the connection between the Java program (JDBC and MySQL) by passing in the proper strings into its getConnection method.

The Connection class was imported from java.sql to allow my Java program and MySQL to be connected (JDBC and MySQL) by also using the DriverManager class to establish a connection. I.e.: Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/***", "***", "***");

The SQLException class was imported from java.sql to make try-catch statements when attempting to do things such as connect the JDBC to MySQL, or run a query, etc. This would tell the program something is wrong if something did in fact go wrong in the program.

PreparedStatement was imported from java.sql for the purpose of executing querys through the use of a Connection object, and its prepareStatement method. This would allow tables in the MySQL database to be changed/ created (all in all, executing querys that contained things such as DDL and SQL statements).

Statement was imported from java.sql so that looping could be done thorough existing tables in the MySQL database. This was done by using the Statement object and then the executeQuery method of the Statement class to perform the action returned by the Connection method createStatement.

The ResultSet class was imported from java.sql to collect data directly from tables in the database in MySQL. The getString and getInt methods were primarily used in order to get specific values from rows that were in a column that was of a type of string (such as collecting grades, which were present in a varchar column).

The ArrayList class was imported so that ArrayLists could be used for the intersectMyShapes and drawIntersectMyShapes method in the MyShapeInterface (used to store MyPoint instances that resided in both shapes).

The HashMap class was imported because HashMap instances were used multiple times throughout the project for the MyPieChart class and the HistogramAlphaBet class.

The Map class was imported because it was used to manipulate HashMaps through the use of the entrySet method (used to iterate through the HashMaps' entries in the program), as well as the getKey and getValue methods to get the keys and values of entries. Additionally, the remove method was used and was taken from this class to remove entries in a HashMap that had a certain key.

The Optional class was used to use the Optional.ofNullable method and the .orElse method when setting values of MyColor in instances to ensure that the variable ‘color’ in all instances were being properly set to real enum MyColor constants.

The Random class was imported to allow my program to fill in the tables with randomly generated values (for emails, genders, etc.). I put the possible genders into an array and made the program choose a random index by the use of a random object and/or the nextInt random method.

The JavaFX Application class was imported because it is the class from which the JavaFX can produce a stage and scene, as well as launch JavaFX in the first place. It also explains the “extends” word for the public class App at the run of the program.

The Scene class of JavaFX was imported because it set the Scene for the canvas to be on in the first place in Part 2 of the project. The scene holds all content in the scene graph and essentially holds all the parts of the JavaFX application.

The Canvas class of JavaFX was imported because that is where the shape instances were all drawn out. The Canvas class allows for drawing to happen on the JavaFX scene.

The GraphicsContext class of JavaFX was imported because it worked directly with the Canvas class and provided the program with information about the canvas so that Part 2 would end up working smoothly. Additionally, it allowed the shapes to be drawn out on the canvas with the use of methods setFill, fillOval, fillRect, setStroke, setLineWidth, and strokeLine.

The Button class of JavaFX was imported because a button was needed to create a GUI in the final output of the project. Additionally, the setOnAction method was used to set what the button was to do when clicked.

The VBox class was imported so that a VBox could be created to hold all the parts of the main program together when everything was to be displayed, those being the Button, Label, and TextField, and at the end, the canvas.

The Color class of JavaFX was imported because it had to work in accordance with the MyColor class I created so that setFill and setStroke would have proper JavaFX colors (which was retrieved through the use of the Get_JavaFXColor of the MyColor class).

The ArcType class of JavaFX was imported because it was needed for the fillArc method in the draw method of the MyArc class (the method needed to know the way the ArcType was to be drawn and this class provides the constants for it). My class Slice also used this since it was essentially linked to the MyArc class.

The Stage class of JavaFX was imported because it provides the main platform to show the results of the code (https://docs.oracle.com/javase/8/javafx/api/javafx/stage/Stage.html). It also was used to use the method setScene to bring up the Scene instance 'scene' on the Stage in the outcome of Part 2. Also the show method was used from the Stage class to reveal the geometric configuration finally produced.

NOTE: Although not directly imported, I had used the Math class of Java (which is in the java.lang package and does not need to be imported - https://www.knowprogram.com/java/import-math-class-java/) for methods pow, sqrt, constant PI, tan, atan, and atan2.