

DMDD ASSIGNMENT 4 - NORMALIZATION

TOPIC NAME: University Recommendation System

Github Repository:- https://github.com/Muskansri1/University_Recommendation_System

Members:

- ❖ Muskan Srivastava (002794929)
- ❖ Sameer Nimse (002752914)

Steps to Ensure 1NF, 2NF, 3NF:

First normal form (1NF)

- Each table has a primary key: minimal set of attributes which can uniquely identify a record
- The values in each column of a table are atomic (No multi-value attributes allowed).
- There are no repeating groups: two columns do not store similar information in the same table

Second normal form (2NF)

- All requirements for 1st NF must be met.
- No partial dependencies.
- No calculated data

Third normal form (3NF)

- All requirements for 2nd NF must be met.
- Eliminate fields that do not directly depend on the primary key; that is no transitive dependencies.

- Primary key for all the tables have been applied.
- All the multi-valued rows have been eliminated/split.
- All the columns store unique information in a table
- All the partial dependencies have been removed by splitting the tables into smaller tables.
- No calculated values have been added.
- All the fields that do not depend directly on the primary key has been removed by splitting the tables into smaller tables.

Below are the screenshots to show the steps taken:

1. Creating New Tables for Recruiter_Details and University_Details since University and Recruiter tables were not following the rules of 2NF and 3NF, hence we decided to split the table.

```

In [1]: import pandas as pd

In [157]: import pymysql
import mysql.connector
uni_recomm = pymysql.connect(host='localhost', user='root', passwd='RootPassword123', database='university_recommend_assignm
cursor= uni_recomm.cursor()

In [162]: cursor.execute("CREATE TABLE University_Details (University_ID varchar(255), University_Name varchar(255),PRIMARY KEY (University
Out[162]: 0

In [163]: cursor.execute("describe University_Details")
for x in cursor:
    print(x)

('University_ID', 'varchar(255)', 'NO', 'PRI', None, '')
('University_Name', 'varchar(255)', 'YES', '', None, '')

In [ ]:

In [164]: UniversityDetails_df=pd.read_csv("University.csv")
UniversityDetails_df

In [174]: cursor.execute("CREATE TABLE Recruiter_Details (Recruiter_ID varchar(255), Recruiter_Name varchar(255),PRIMARY KEY (Recruiter_ID)
Out[174]: 0

In [224]: cursor.execute("describe Recruiter_Details")
for x in cursor:
    x

```

2. Inserting into University_Details table

```

In [164]: UniversityDetails_df=pd.read_csv("University.csv")
UniversityDetails_df

```

		Gainesville												
40	42	University of Houston, Clear Lake	34260.0	On-Campus	1.8	Fall	2023	MSCS	Clear Lake	9460.0	31482	290	79	6.0
41	43	University of Illinois, Urbana Cham...	70100.0	On-Campus	1.8	Fall	2023	MSCS	Urbana champaign	6777.0	36798	Waived	103	7.5
42	44	University of Maryland	680000.0	On-Campus	2.0	Fall	2023	MSCS	College Park	53036.0	26535	320	95	7.0
43	45	University of Michigan, Ann Arbor	75842.0	On-Campus	1.0	Fall	2022	MSCSE	Ann Arbor	75410.0	100470	Waived	84	6.5
44	46	University of Michigan, Dearborn	75842.0	On-Campus	1.8	Fall	2022	DSBI	Dearborn	28786.0	46380	Waived	84	6.5
45	47	University of Minnesota, Twin	72300.0	On-Campus	2.0	Fall	2022	DSBI	Twin cities	17009.0	40766	Waived	79	6.5

```

In [165]: for i,row in UniversityDetails_df.iterrows():
            cursor.execute("INSERT INTO University_Details values (%s,%s)", (int(row['University_ID']),row['University_Name']))
            uni_recomm.commit()

            cursor.execute("SELECT * from University_Details")
            records=cursor.fetchall()

            print(records)
            uni_recomm.commit()

```

3. Removing the Column University_Name from the University Table as we considered this value in the University_Details Table.

```
In [166]: cursor.execute("select * from university")
for x in cursor:
    print(x)

('0', '306', '79', '6.5')
('28', 'State University of New York, Buffalo', 'None', 'On-Campus', '2', 'Fall', 2023, 'MSCS', 'New York', 'None', '46200', '306', '79', '6.5')
('29', 'State University of New York, Polytechnic Institute', '83000', 'On-Campus', '1.8', 'Fall', 2022, 'MTM', 'New York', '15287', '23112', 'Waived', '79', '7.5')
('3', 'Boston University', '80000', 'On-Campus', '2', 'Fall', 2022, 'MSDS', 'Boston', '9600', '90000', '300', '110', '8')
('30', 'State University of New York, Stony Brook', 'None', 'On-Campus', '1.8', 'Fall', 2022, 'MSCS', 'New York', 'None', '29853', '300', '80', '6.5')
('31', 'Texas A&M University, College Station', '175000', 'On-Campus', '1.8', 'Fall', 2022, 'MSCS', 'Texas', '21950', '24940', '295', '80', '6')
('32', 'Texas A&M University, Kingsville', '175000', 'On-Campus', '2', 'Fall', 2023, 'MSCS', 'Texas', '40754', '23908', '297', '79', '6')
('33', 'University of California, Irvine', '109000', 'On-Campus', '1.5', 'Fall', 2022, 'MSCS', 'Irvine', '77431', '33502', '302', '80', '7')
('34', 'University of California, Los Angeles', '64433', 'On-Campus', '1.5', 'Fall', 2022, 'MSBA', 'Los Angeles', '20000', '14625', 'None', '87', '7')
('35', 'University of California, San Diego', '77143', 'On-Campus', '2', 'Fall', 2022, 'MSCSE', 'San Diego', '52242', '22884', 'None', '83', '7')
('36', 'University of Colorado, Boulder', '43000', 'On-Campus', '1.8', 'Fall', 2022, 'MSCS', 'Boulder', '20000', '53736', 'Waived', '83', '6')

In [167]: cursor.execute("ALTER Table University drop column University_Name")
for x in cursor:
    print(x)

In [168]: cursor.execute("select * from university")
for x in cursor:
    print(x)

('1', '100000', 'On-Campus', '2', 'Fall', 2022, 'MSIS', 'Boston', '9600', '54000', '310', '100', '7.5')
('10', '36100', 'On-Campus', '1.8', 'Fall', 2022, 'MSIS', 'Cleveland', '24693', '21702', '295', '78', '6')
('11', '44700', 'On-Campus', '2', 'Fall', 2022, 'MSIS', 'Boston', '9600', '54000', '310', '100', '7.5')
```

4. Removing the Column University_Name from the Student Table since University_Detail table would hold the value and University_ID has been referenced as foreign key from University_Details table to Student_Table

```
In [170]: cursor.execute("ALTER Table Student drop column University_Name")
for x in cursor:
    print(x)
```

5. Inserting into Recruiter_Details Table and deleting Recruiter_Name from Recruiter table

```
] for i,row in RecruiterDetails_df.iterrows():
    cursor.execute("INSERT INTO recruiter_details values (%s,%s)", (int(row['Recruiter_ID']),row['Recruiter_Name']))
    uni_recomm.commit()

cursor.execute("SELECT * from recruiter_details")
records=cursor.fetchall()

print(records)
uni_recomm.commit()

((('1', 'Cognizant Technology Solutions'), ('10', 'Hcl America'), ('11', 'Accenture'), ('12', 'IBM'), ('13', 'Deloitte Consulting'), ('14', 'Intel'), ('15', 'Capgemini'), ('16', 'Wal-Mart Associates'), ('17', 'Jpmorgan Chase'), ('18', 'Wipro'), ('19', 'Teksorg'), ('2', 'Amazon'), ('20', 'Compunnel Software Group'), ('21', 'Qualcomm Technologies'), ('22', 'Salesforce.Com'), ('23', 'Tech Mahindra'), ('24', 'Cisco Systems'), ('25', 'Goldman Sachs &'), ('3', 'Tata Consultancy Services'), ('4', 'Google'), ('5', 'Microsoft'), ('6', 'Facebook'), ('7', 'Ernst & Young'), ('8', 'Infosys'), ('9', 'Apple'))

]) cursor.execute("ALTER Table recruiter drop column Recruiter_Name")
for x in cursor:
    print(x)

]) cursor.execute("select * from recruiter")
for x in cursor:
    print(x)

('1', 1, 12681, '$92,766', '32')
('10', 10, 4113, '$99,754', '25')
('11', 11, 3830, '$140,893', '20')
('12', 12, 3800, '$126,521', '59')
('13', 13, 3756, '$119,341', '41')
('14', 14, 3367, '$127,627', '26')
('15', 15, 3325, '$102,410', '1')
```

6. Professor table had multi-valued rows which broke the rules of 2NF and 3NF

```
NewProfessor_df=pd.read_csv("Professor_Final.csv")
NewProfessor_df
```

	ID	University_ID	Professors
0	1	58	Pieter Abbeel, Rediet Abebe, Ahmed Alaa, Elad ...
1	2	36	Titan Alon,James Andreoni, Kate Antonovics, Da...
2	3	2	Jacob Abraham, Maruthi Akella, Deji Akinwande,...
3	4	26	Elizabeth Reese, Paula Moya, Heike Daldrup-Lin...
4	5	42	Abadi Daniel J., Abbasi Hossein, Abed Eyad, Ab...
5	6	41	Butler Wingfield Kim, Butts Daniel, Cable'tJoh...
6	7	48	Victor Adamchik, Leonard M Adleman, Murali Ann...
7	8	6	Abbaszadegan Morteza, Abbott Joshua, Aberle Ja...
8	9	1	Mehdi Abedi, Emad Aboelela, Gregory D. Abowd, ...
9	10	61	S. Sebnem Ahiska King, Karen Chen, Jingyan Don...
10	11	62	Lo'ay Abusalah, Rashid Ansari, Jezekiel Ben-Ar...
11	12	63	Patricia A. Young, Zane L. Berge, Francis M. H...
12	13	47	Scott Solomon, Cheryl Hall, David Ponton, Nico...
13	14	51	Abbas Waseem, Ackerman Robert, Adams Stephanie...
14	15	4	Keith Alford, Nick Armstrong, Kristen Aust, Th...
15	16	64	ELIZABETH ANDERSON, CHARLOTTE ADAMS, ALYSSA AN...
16	17	23	Ali Tosyali, Edwin N. Torres, Rick Mislan, Mik...

7. Splitting the values to remove multi-valued rows:

```
In [185]: newProfessorsList_df = pd.DataFrame({
          :   col:np.repeat(x[col].values, x[1st_col].str.len())
          :   for col in x.columns.difference([1st_col])
          :   }).assign(**{1st_col:np.concatenate(x[1st_col].values)})[x.columns.tolist()]
```

```
In [186]: newProfessorsList_df
```

```
Out[186]:
```

	ID	University_ID	Professors
0	1	58	Pieter Abbeel
1	1	58	Rediet Abebe
2	1	58	Ahmed Alaa
3	1	58	Elad Alon
4	1	58	Venkat Anantharam
...
192	17	23	Joy Olabisi
193	17	23	Clyde Eirikur Hull
194	17	23	Raj Murthy
195	17	23	Mike Palanski
196	17	23	

197 rows x 3 columns

8. Similarly, removing the multi-valued rows from research and splitting them into separate rows:

```
In [185]: newProfessorsList_df = pd.DataFrame({
    col:np.repeat(x[col].values, x[1st_col].str.len())
    for col in x.columns.difference([1st_col])
}).assign(**{1st_col:np.concatenate(x[1st_col].values)})(x.columns.tolist())
```

```
In [186]: newProfessorsList_df
```

```
Out[186]:
```

	ID	University_ID	Professors
0	1	58	Pieter Abbeel
1	1	58	Rediet Abebe
2	1	58	Ahmed Alaa
3	1	58	Elad Alon
4	1	58	Venkat Anantharam
...
192	17	23	Joy Olabisi
193	17	23	Clyde Eirikur Hull
194	17	23	Raj Murthy
195	17	23	Mike Palanski
196	17	23	

197 rows × 3 columns

Research_ID	Research_Name	University_ID	
0	1	Environmental Economics and Policy	15
1	2	Investigating the Mold Resiliency of Building...	15
2	3	Investigating Energy Efficiency and Heat Vuln...	15
3	4	Research and Fact-Checking for Book Chapter a...	15
4	5	Aging and Climate Change in China	15
5	6	Modeling of Atmospheric Chemistry	15
6	7	Climate Dynamics	15
7	8	Fulton Undergraduate Research Initiative (FURI)	6
8	9	New College Undergraduate Inquiry and Researc...	6
9	10	School of Life Sciences Undergraduate Researc...	6
10	11	Sustainability Undergraduate Research Experie...	6
11	12	Watts College of Public Service and Community...	6
12	13	Ira A. Fulton Schools of Engineering Summer ...	6
13	14	Web & Visualization Developer (Media Cloud Pro...	1
14	15	Research in Programming Languages and Machine...	1
15	16	Multiple research options on experimental mul...	1
16	17	Imaging microbiome interactions: Microfluidics	1
17	18	Microscopy	1
18	19	Molecular Rinkov	1

9. Removing Multi-valued rows from Student Table:

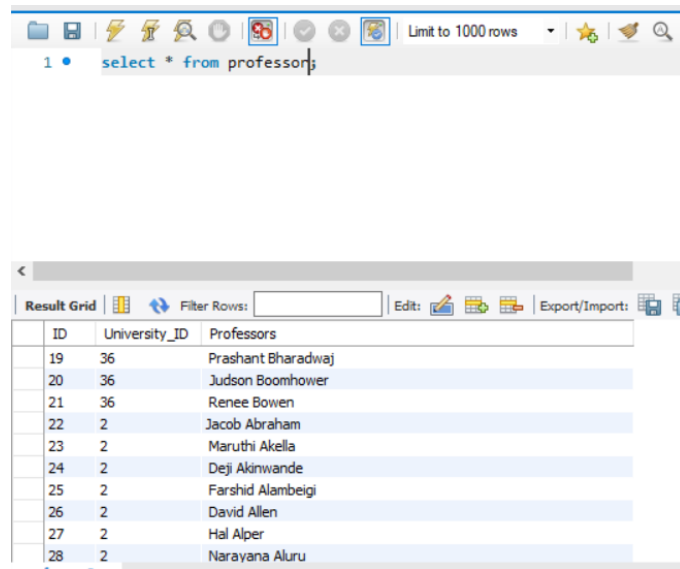
Admit', 'MS', 'Computer Science', 'Fall', '2014', '159', '148', '307', '3', '100', 'VTU', 'Computer Science', '70.1', '10
0', '70'), (78, '1', 'Admit', 'MS', 'Computer Science', 'Fall', '2012', '800', '550', '1350', '3.5', '91', 'VTU', 'Informati
on Science', '73', '100', '0'), (79, '1', 'Admit', 'MS', 'Information Systems', 'Fall', '2011', '720', '520', '1240', '2',
'21', 'MU', 'Computer Engineering', '0', '100', '0'), (80, '1', 'Admit', 'MS', 'Industrial Engineering', 'Fall', '2015', '17
0', '152', '322', '3.5', '98', 'Guru Gobind Singh Indraprastha University', 'TOOL ENGINEERING', '79.47', '100', '18'), (81,
'1', 'Admit', 'MS', '(MIS / MSIM / MSIS / MSIT)', 'Fall', '2015', '156', '148', '304', '3.5', '100', 'MU', 'Electronics',
'0', '0', '0'), (82, '1', 'Admit', 'MS', 'Computer Science', 'Fall', '2012', '160', '152', '312', '3.5', '108', 'JNTU', 'EC
E', '7.6', '10', '0'), (83, '1', 'Admit', 'MS', 'Computer Science', 'Fall', '2012', '800', '590', '1390', '4', '109', 'Manip
al Institute of Technology', 'Computer Science and Engineering', '7.71', '10', '0'), (84, '1', 'Admit', 'MS', 'Computer Scienc
e', 'Fall', '2012', '156', '152', '308', '3.5', '112', 'Jai Narain Vyas University Jodhpur', 'Computer Science and Engineerin
g', '73.4', '100', '0'), (85, '1', 'Admit', 'MS', 'Industrial Engineering', 'Fall', '2013', '162', '155', '317', '3.5', '11
3', 'GGSIU', 'Mechanical', '70', '100', '0'), (86, '1', 'Admit', 'MS', 'MIS', 'Fall', '2011', '740', '490', '1230', '3', '1
01', 'MU', 'I.T', '64', '100', '0'), (87, '1', 'Admit', 'MS', 'Computer Science', 'Fall', '2013', '163', '154', '317', '3',
'103', 'ITM Gurgaon', 'CSE', '64.3', '100', '0'), (88, '1', 'Admit', 'MS', 'Computer Science', 'Fall', '2011', '780', '550',
'1330', '4.5', '107', 'K J Somaiya College of Engineering', 'Information Technology', '71.4', '100', '0'), (89, '1', 'Admi
t', 'MS', 'Computer Science', 'Fall', '2014', '161', '153', '314', '3', '109', 'Shivaji University', 'Electronics Engineerin
g', '70.3', '100', '0'), (90, '1', 'Admit', 'MS', 'Engineering Management', 'Spring', '2011', '700', '530', '1230', '4', '11
0', 'Sinhgad College of Engineering', 'Chemical Engineering', '60', '100', '0'), (91, '1', 'Admit', 'MS', 'Computer Science',
'Fall', '2012', '800', '650', '1450', '4', '110', 'Vidyalankar Institute of Technology', 'Information Technology', '71.46',
'100', '0'), (92, '1', 'Admit', 'MS', 'Mechanical Engineering', 'Fall', '2014', '155', '149', '304', '3', '101', 'BBDT Ghaz

Out[217]: 467

```
1 • select * from course;
```

ID	University_ID	Course_Name
1	1	3-D Animation
2	1	Accounting
3	1	Accounting and Business Administration (MS/MBA)
4	1	Accounting and Financial Decision Making
5	1	Adult-Gerontology Acute Care Nursing
6	1	Adult-Gerontology Acute Care Nursing (CAGS)
7	1	Adult-Gerontology Primary Care Nursing (CAGS)
8	1	Advanced and Intelligent Manufacturing
9	1	Agile Project Management
10	1	Analytics

2.

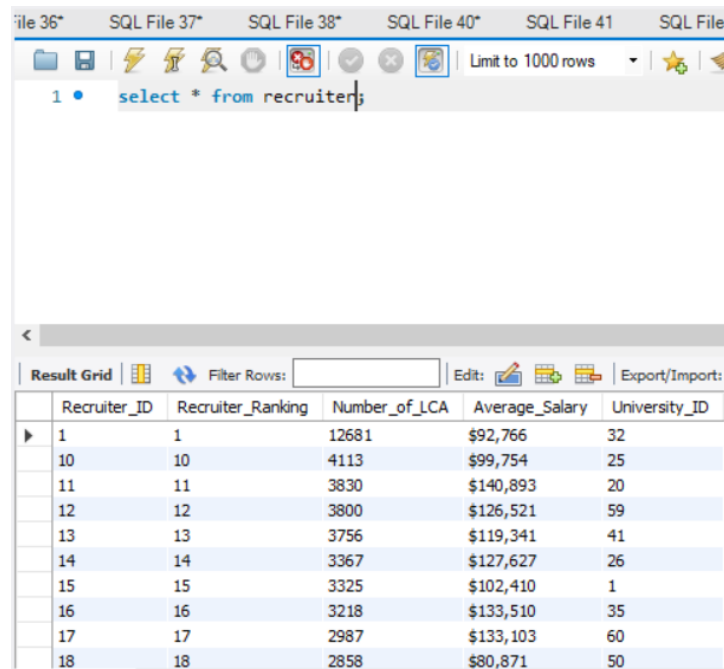


SQL Developer interface showing the query: `select * from professors;`

Result Grid:

ID	University_ID	Professors
19	36	Prashant Bharadwaj
20	36	Judson Boomhower
21	36	Renee Bowen
22	2	Jacob Abraham
23	2	Maruthi Akella
24	2	Deji Akinwande
25	2	Farshid Alambeigi
26	2	David Allen
27	2	Hal Alper
28	2	Narayana Aluru

3.



SQL Developer interface showing the query: `select * from recruiter;`

Result Grid:

Recruiter_ID	Recruiter_Ranking	Number_of_LCA	Average_Salary	University_ID
1	1	12681	\$92,766	32
10	10	4113	\$99,754	25
11	11	3830	\$140,893	20
12	12	3800	\$126,521	59
13	13	3756	\$119,341	41
14	14	3367	\$127,627	26
15	15	3325	\$102,410	1
16	16	3218	\$133,510	35
17	17	2987	\$133,103	60
18	18	2858	\$80,871	50

4.

File 36* SQL File 37* SQL File 38* SQL File 40* SQL File 41

Limit to 1000 rows

```
1 • select * from recruiter_details;
```

Result Grid

Recruiter_ID	Recruiter_Name
1	Cognizant Technology Solutions
10	Hd America
11	Accenture
12	IBM
13	Deloitte Consulting
14	Intel
15	Capgemini
16	Wal-Mart Associates
17	Jpmorgan Chase
18	Winn

5.

File 36* SQL File 37* SQL File 38* SQL File 40* SQL File 41 SQL File 42* SQL File 43*

Limit to 1000 rows

```
1 • select * from research;
```

Result Grid

Research_ID	Research_Name	University_ID
1	Environmental Economics and Policy	15
10	School of Life Sciences Undergraduate Research (SOLUR)	6
11	Sustainability Undergraduate Research Experience (SURE)	6
12	Watts College of Public Service and Community Solutions Undergraduate Res...	6
13	Ira A. Fulton Schools of Engineering Summer Research Internships (SURI)	6
14	Web & Visualization Developer (Media Cloud Project)	1
15	Research in Programming Languages and Machine Learning	1
16	Multiple research options on experimental multiphase flow with potential appli...	1
17	Imaging microbiome interactions: Microfluidics	1
18	Microscopy	1
19	Molecular Biology	1
2	Investigating the Mold Resiliency of Buildings in Future Climates	15
20	and Automation	1

6.

File 36* SQL File 37* SQL File 38* SQL File 40* SQL File 41 SQL File 42* SQL File 43*

Limit to 1000 rows

```
1 • select * from scholarship;
```

Result Grid

	Scholarship_ID	Scholarship_Name	Scholarship_URL
▶	1	Cornaro Scholarship	https://www.kappagammapi.org/cornaro-schol...
	2	NCAA Postgraduate Scholarship Program	http://www.ncaa.org/ncaa-postgraduate-schol...
	3	Eiffel Excellence Scholarship Program	https://www.campusfrance.org/en/eiffel-schola...
	4	Fulbright Foreign Student Program	https://us.fulbrightonline.org/
	5	Clarendon Scholarships at University of Oxford	http://www.ox.ac.uk/clarendon/
	6	Gates Cambridge Scholarship	https://www.gatescambridge.org/
	7	Kate Neal Kinley Memorial Fellowship	http://faa.illinois.edu/alumni-friends/kate-neal-k...
	8	Forte Fellows Program	http://www.fortefoundation.org/site/PageServ...
	9	Richard A. Freund International Scholarship	https://asq.org/about-asq/asq-awards/freunds...
	10	Olivia James Traveling Fellowship	https://www.archaeological.org/grant/james-tr...
	11	Helen M. Woodruff Fellowship	https://www.archaeological.org/grant/woodruf...
	12	Anna C. and Oliver C. Colburn Fellowship	https://www.archaeological.org/grant/colburn-f...
	13	Harriet and Leon Pomerance Fellowship	https://www.archaeological.org/grant/pomeran...

7.

File 36* SQL File 37* SQL File 38* SQL File 40* SQL File 41 SQL File 42* SQL File 43* SQL File

Limit to 1000 rows

```
1 • select * from student;
```

Result Grid

	student_id	University_ID	Acceptance	Program	Target_Major	Term	Year_of_Joining	GRE_Q
▶	1	1	Admit	MS	Computer Science	Fall	2012	157
	2	1	Admit	MS	Industrial Engineering	Fall	2011	None
	3	1	Admit	MS	Industrial Engineering	Fall	2011	770
	4	1	Admit	MS	Computer Science	Fall	2013	159
	5	1	Admit	MS	Management Information System	Fall	2014	158
	6	1	Admit	MS	Computer Science	Fall	2013	162
	7	1	Admit	MS	Computer Science	Fall	2013	160
	10	1	Admit	MS	MIS	Fall	2011	760
	11	1	Admit	MS	Information Security	Fall	2009	740
	13	1	Admit	MS	Electronics and Communication	Fall	2012	780
	15	1	Admit	MS	Information Security	Fall	2013	158
	16	1	Admit	MS	Computer Science	Fall	2012	800

File 36* SQL File 37* SQL File 38* SQL File 40* SQL File 41 SQL File 42* SQL File 43* SQL File 44* SQL File 45* SQL File 46* SQL File

Limit to 1000 rows

```
1 • select * from student;
```

Result Grid

	sr_of_Joining	GRE_Q	GRE_V	GRE_Total	GRE_AWA	TOEFL	Undergrad_University	Undergrad_Major	CGPA	CGPA_Scale	Work_Ex
▶	12	157	142	299	3.5	97	University of Mumbai	Computer Engineering	64.37	100	0
	11	None	None	0	None	None			0	0	0
	11	770	460	1230	3	105	fr.c.r.c.e.(bandra)	production engineering	75.36	100	0
	13	159	152	311	3	94	None	CS	3.3	10	0
	14	158	144	302	3	100	Apeejay College of Engineering	ECE	66.4	100	0
	13	162	155	317	4	None	RNSIT	CS	72	100	0
	13	160	154	314	3.5	108	Sri Bhagawan Mahaveer Jain College of Engine...	Information Science	73	100	0
	11	760	680	1440	4	111	Fr CRIT Mumbai University	Mechanical Engineering	54.6	100	0
	19	740	540	1280	5	113	VTU	CS	67	100	0
	12	780	590	1370	3.5	114	Manipal Institute of Technology	ECE	8.3	10	0
	13	158	148	306	3	94	Rajasthan Technical University	Computer Engineering	67	100	0
	12	800	610	1410	3.5	110	UPTU	CSE	74	100	0

8.

file 36* SQL File 37* SQL File 38* SQL File 40* SQL File 41 SQL File 42* SQL File 43* SQL File 44* SQL File 45* SQL File 46*

Limit to 1000 rows

1 • select * from university;

Result Grid

	University_ID	avg_pay_scale	Delivery_Mode	Duration	Term	Year_of_Joining	Major	Location	Cost_of_Living	Avg_Fees	GRE	toefl	ielts
▶	1	100000	On-Campus	2	Fall	2022	MSIS	Boston	9600	54000	310	100	7.5
	10	36100	On-Campus	1.8	Fall	2022	MSIS	Cleveland	24693	21702	295	78	6
	11	113700	On-Campus	2	Fall	2022	MSCS	Ithaca	19726	29500	295	100	7
	12	24249	On-Campus	2	Fall	2022	MSCS	Canada	16776	16860	300	92	7
	13	56200	On-Campus	2	Fall	2022	MSCS	Virginia	59782	22110	Waived	88	7
	14	94000	On-Campus	2	Fall	2022	MSCS	Atlanta	55672	18225	Waived	79	6.5
	15	128900	On-Campus	1.8	Fall	2022	MSCS	Boston	83463	89952	Waived	80	6.5
	16	68647	On-Campus	2	Fall	2022	MSCS	Chicago	12600	26336	309	90	6.5
	17	136000	On-Campus	1.8	Fall	2023	MSCS	Indiana	38680	24375	Waived	100	7
	18	60156	On-Campus	2	Fall	2022	MSCS	Iowa	35004	23790	316	79	6.5
	19	45700	On-Campus	2	Fall	2023	MSCS	Kansas	15046	14385	310	79	6.5
	2	90000	On-Campus	2	Fall	2022	MSCS	Texas	7800	60000	300	100	7.5
	20	83000	On-Campus	2	Fall	2022	MSCS	New York	40000	35982	321	100	7

9.

file 36* SQL File 37* SQL File 38* SQL File 40* SQL File 41

Limit to 1000 rows

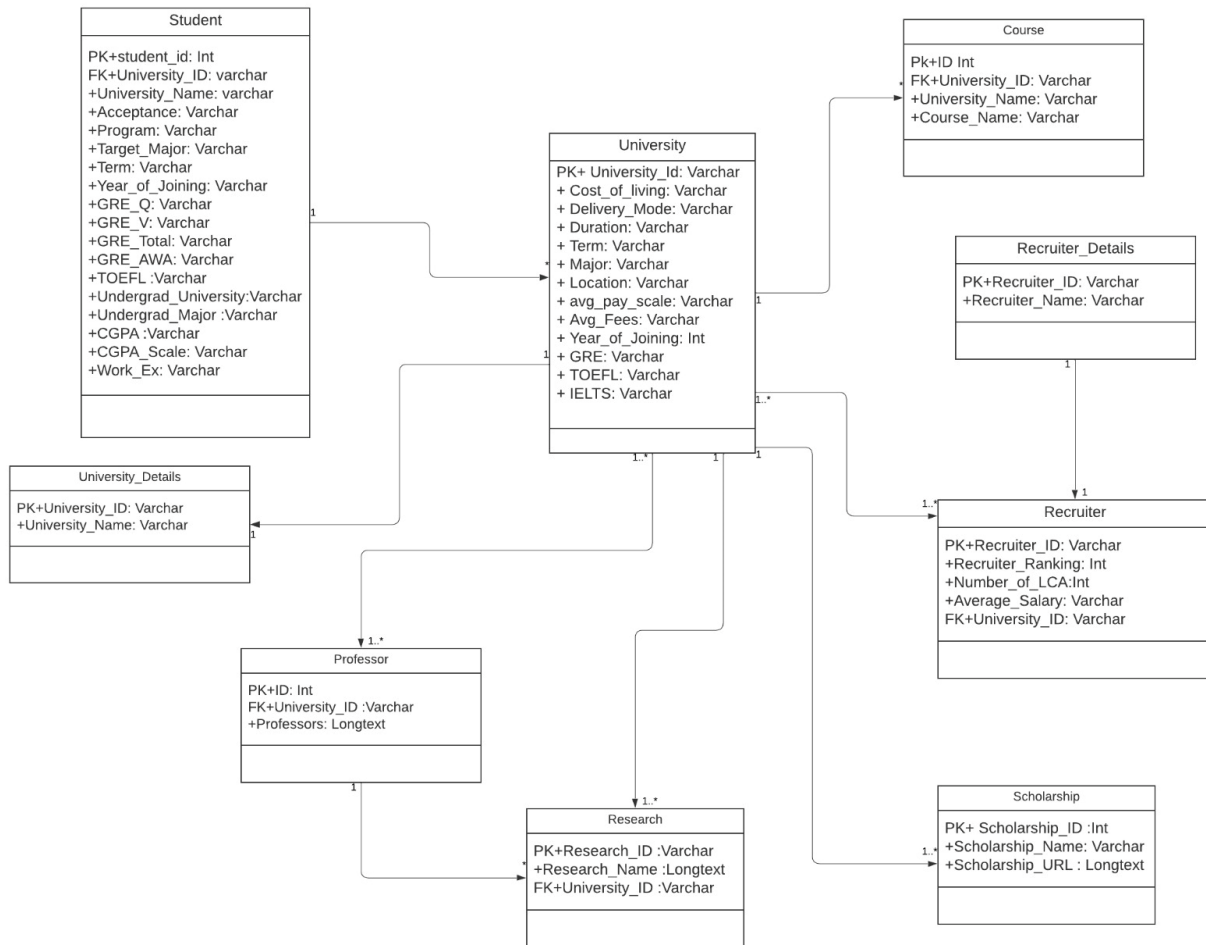
1 • select * from university_details;

Result Grid

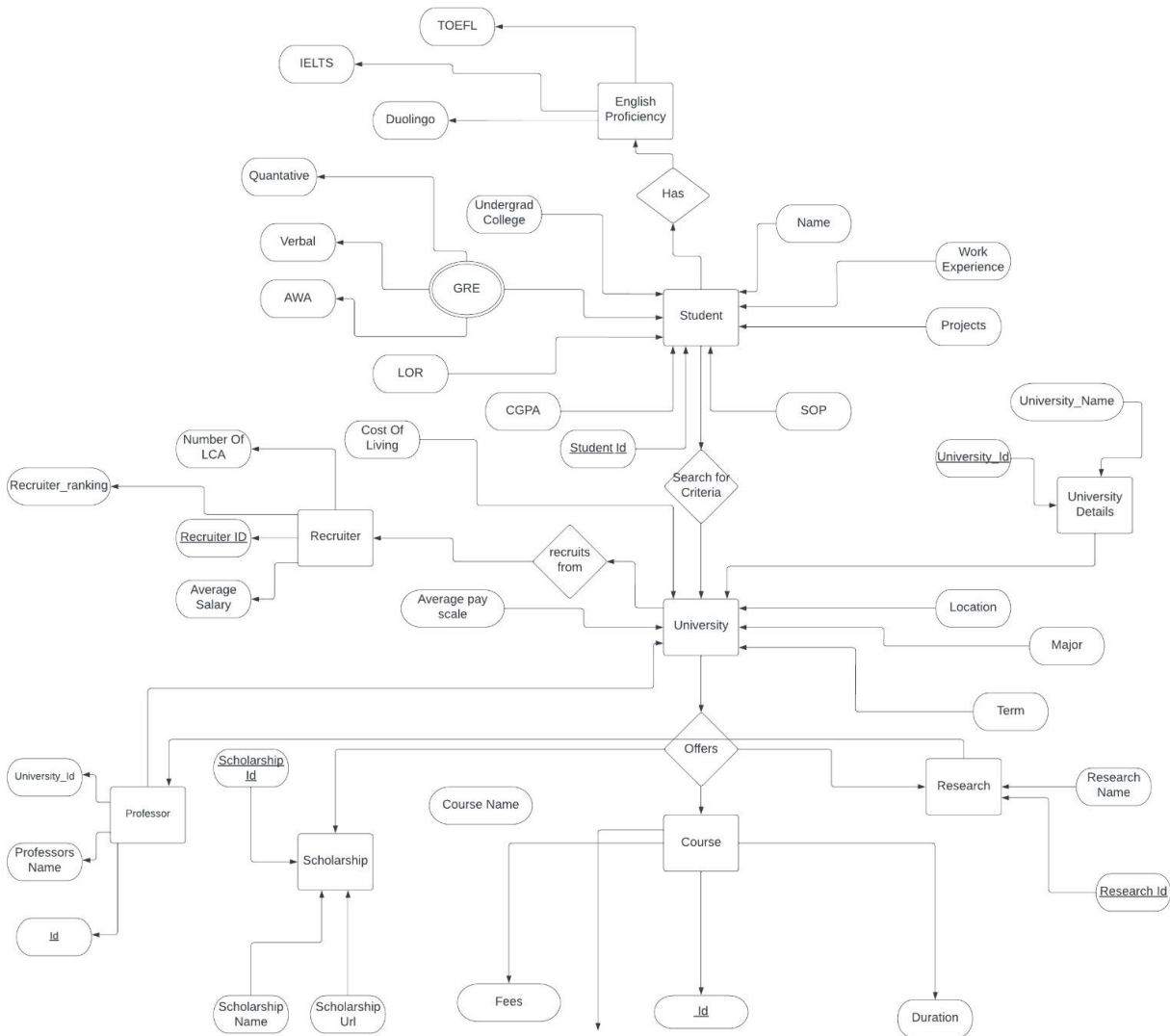
	University_ID	University_Name
▶	1	Northeastern University
	10	Cleveland State University
	11	Cornell University
	12	Dalhousie University
	13	George Mason University
	14	Georgia Institute of Technology
	15	Harvard University
	16	Illinois Institute of Technology, Chicago
	17	Indiana University, Bloomington
	18	Iowa State University
	19	Kansas State University
	2	University of Texas
	21	New York University

ER Diagram and Class Diagram

CLASS DIAGRAM UNIVERSITY RECOMMENDATION SYSTEM



ER Diagram University Recommendation System



USE CASES USING VIEWS:-

1. **Search for a distinct research opportunity (for example – Machine Learning) available in a particular university along with the list of all the professors of that university.**

Description: This query returns a list of ongoing research in a university under a professor

Actor: Student

Precondition: The student must select research opportunities

Steps:

Actor Action: The student views the ongoing research from a university

System Response: The system displays all the ongoing research from a university under the professor

Postcondition: The system generates a list of ongoing research

SQL Query

create view Distinct_Research_Opportunity as

select distinct(r.Research_Name), u.University_Name,p.professors from
professor p

inner join university_details u on p.university_id = u.university_id

inner join research r on r.university_id = u.university_id

where r.Research_Name like '%Learning%';

The screenshot shows a database management interface with a left sidebar for 'SCHEMAS' and a main area for SQL queries. The 'SCHEMAS' sidebar lists databases like 'sakila', 'sys', and 'unibuddydemo'. The 'unibuddydemo' database is expanded, showing tables, views, stored procedures, and functions. The main area displays an SQL query in a text editor, which is the same query provided in the text above. Below the editor, a 'Result Grid' shows the output of the query. The grid has three columns: 'Research_Name', 'University_Name', and 'professors'. The data is as follows:

Research_Name	University_Name	professors
Research in Programming Languages and Machine Learning	Northeastern University	Kal Bugara
Research in Programming Languages and Machine Learning	Northeastern University	Vishal Chavla
Research in Programming Languages and Machine Learning	Northeastern University	Nick Brown
Research in Programming Languages and Machine Learning	Northeastern University	Daniel Peters
Research Experience and Applied Learning (REAL) Portal	University of California, Irvine	Titan Alon
Research Experience and Applied Learning (REAL) Portal	University of California, Irvine	James Andreoni
Research Experience and Applied Learning (REAL) Portal	University of California, Irvine	Kate Antonovics
Research Experience and Applied Learning (REAL) Portal	University of California, Irvine	David Arnold

2. Display the recruiter ranking along with the average salary and the university from which it hires the maximum number of students

Description: Display a list of recruiters along with the average salary that they offer and the maximum number of students they hire from a particular university

Actor: Student

Precondition: The student must select a list of top recruiters

Steps:

Actor Action: The student views the recruiters along with the average package

System Response: The system displays a list of all the recruiters and the university from which they hire the maximum number of students

Postcondition: The system generates a list of recruiters

SQL Query:

create view Recruiter_Ranking as

```
select rn.Recruiter_Name, r.Recruiter_Ranking, r.Number_of_LCA,  
r.Average_Salary, u.university_name as Max_Hiring_from_University  
from recruiter r, university_details u, recruiter_details rn  
where u.University_ID = r.University_ID and rn.Recruiter_ID = R.Recruiter_ID  
order by r.Recruiter_Ranking asc;
```

The screenshot shows a database management tool interface. On the left, a 'Navigator' pane displays a schema tree for 'unibuddydemo'. The main area shows a SQL query editor with the following code:

```
1 • create view Recruiter_Ranking as  
2   select rn.Recruiter_Name, r.Recruiter_Ranking, r.Number_of_LCA, r.Average_Salary, u.university_name as Max_Hiring_from_University  
3   from recruiter r, university_details u, recruiter_details rn  
4   where u.University_ID = r.University_ID and rn.Recruiter_ID = R.Recruiter_ID  
5   order by r.Recruiter_Ranking asc;  
6  
7 • select * from Recruiter_Ranking;
```

Below the query editor, a 'Result Grid' displays the results of the query. The grid has five columns: Recruiter_Name, Recruiter_Ranking, Number_of_LCA, Average_Salary, and Max_Hiring_from_University. The data is as follows:

Recruiter_Name	Recruiter_Ranking	Number_of_LCA	Average_Salary	Max_Hiring_from_University
Cognizant Technology Solutions	1	12681	\$92,766	Syracuse University
Tata Consultancy Services	3	9822	\$93,484	Indiana University, Bloomington
Google	4	9421	\$156,793	San Jose State University
Facebook	6	6090	\$175,750	Western Michigan University
Ernst & Young	7	5797	\$118,653	University of South Carolina, Columbia
Infosys	8	5649	\$92,689	University of Texas, Arlington
Hd America	10	4113	\$99,754	Saint Peter's University
IBM	12	3800	\$176,531	Michigan State University

3. Display the list of all the professors that teach under a particular university.

Description: The student views the professors from a particular university

Actor: Student

Precondition: The student must select a university

Steps:

Actor Action: The student views the professor from a particular university

System Response: The system generates a list of professors from a university

Postcondition: The system generates all the professors

SQL Query:

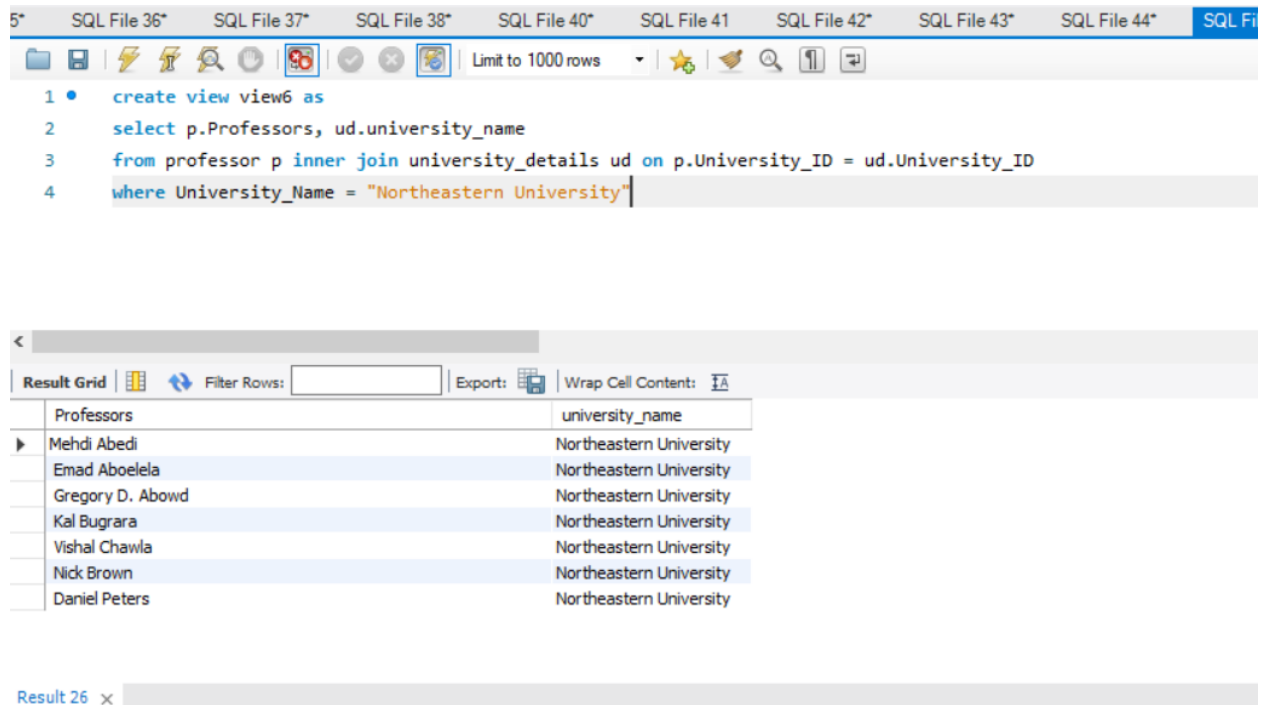
create view view6 as

```
select p.Professors, ud.university_name
```

```

from professor p inner join university_details ud on p.University_ID =
ud.University_ID
where University_Name = "Northeastern University"

```



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```

1 • create view view6 as
2   select p.Professors, ud.university_name
3   from professor p inner join university_details ud on p.University_ID = ud.University_ID
4   where University_Name = "Northeastern University"

```

The results grid displays the output of the query, showing a list of professors and their university names. The grid has two columns: "Professors" and "university_name". The data is as follows:

Professors	university_name
Mehdi Abedi	Northeastern University
Emad Aboelela	Northeastern University
Gregory D. Abowd	Northeastern University
Kal Bugrara	Northeastern University
Vishal Chawla	Northeastern University
Nick Brown	Northeastern University
Daniel Peters	Northeastern University

Below the results grid, there is a tab labeled "Result 26" with a close button (x).

4. Display the list of all the courses offered by an university along with the list of all the professors present in the university.

Description: This query returns a list of professors along with their university ID and the courses offered by that university

Actor: Student

Precondition: The student must select a professor

Steps:

Actor Action: The student views the professor teaching a particular course from a university

System Response: The system displays a list of all the professors from a university along with the professors

Postcondition: A list is generated of all the professors at a university along with their courses

SQL Query:

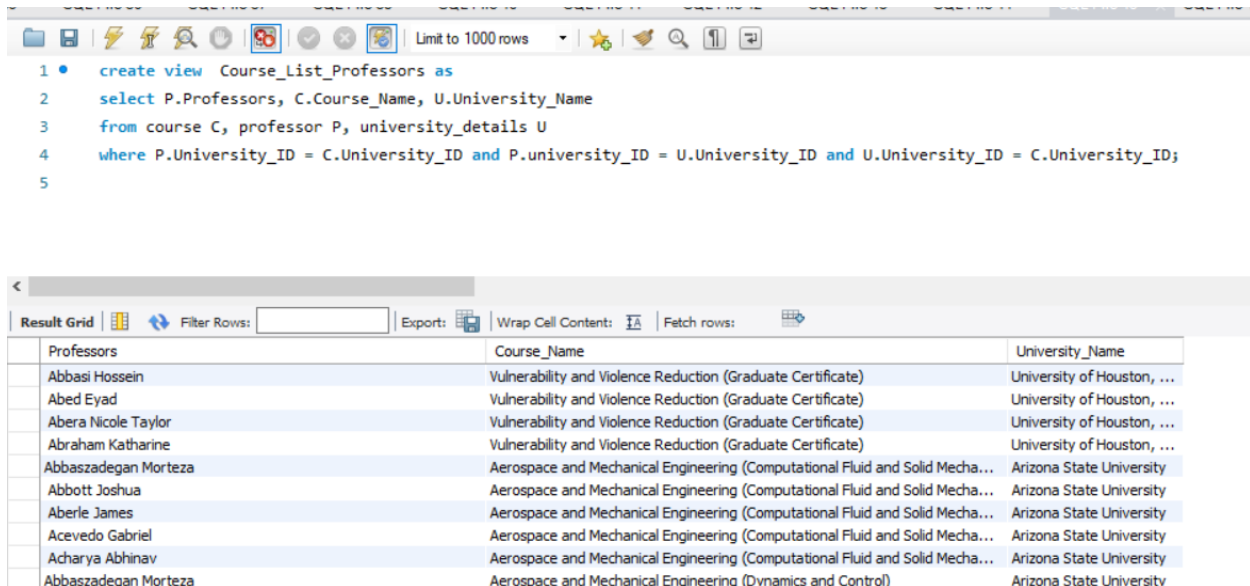
```
create view Course_List_Professors as
```



```

select P.Professors, C.Course_Name, U.University_Name
from course C, professor P, university_details U
where P.University_ID = C.University_ID and P.university_ID = U.University_ID
and U.University_ID = C.University_ID;

```



Limit to 1000 rows

```

1 • create view Course_List_Professors as
2 select P.Professors, C.Course_Name, U.University_Name
3 from course C, professor P, university_details U
4 where P.University_ID = C.University_ID and P.university_ID = U.University_ID and U.University_ID = C.University_ID;
5

```

Professors	Course_Name	University_Name
Abbasi Hossein	Vulnerability and Violence Reduction (Graduate Certificate)	University of Houston, ...
Abed Eyad	Vulnerability and Violence Reduction (Graduate Certificate)	University of Houston, ...
Abera Nicole Taylor	Vulnerability and Violence Reduction (Graduate Certificate)	University of Houston, ...
Abraham Katharine	Vulnerability and Violence Reduction (Graduate Certificate)	University of Houston, ...
Abbaszadegan Morteza	Aerospace and Mechanical Engineering (Computational Fluid and Solid Mecha...	Arizona State University
Abbott Joshua	Aerospace and Mechanical Engineering (Computational Fluid and Solid Mecha...	Arizona State University
Aberle James	Aerospace and Mechanical Engineering (Computational Fluid and Solid Mecha...	Arizona State University
Acevedo Gabriel	Aerospace and Mechanical Engineering (Computational Fluid and Solid Mecha...	Arizona State University
Acharya Abhinav	Aerospace and Mechanical Engineering (Computational Fluid and Solid Mecha...	Arizona State University
Abbaszadegan Morteza	Aerospace and Mechanical Engineering (Dynamics and Control)	Arizona State University

5. Display the conditions on which a student is admitted to a university along with their graduate major, and average pay scale after graduation

Description: Displays the entire academics of a student based on admit

Actor: Student

Precondition: The student must select a student ID

Steps:

Actor Action: The student selects a student ID and views the entire academic

System Response: The system generates a list of all the admitted students

Postcondition: A list is generated with all the amidst

SQL Query:

create view View1 as

```

Select s.student_ID, s.acceptance, s.GRE_Q, s.GRE_V,s.GRE_AWA,s.TOEFL,
s.Undergrad_University, s.Undergrad_Major, u.university_name,
un.avg_pay_scale, r.university_id from Student s, University_Details u, university
un, recruiter r where s.university_id = u.university_id and u.university_id =
r.university_id and un.University_ID = u.University_ID and s.acceptance =
'Admit';

```


SQL File 36* SQL File 37* SQL File 38* SQL File 40* SQL File 41 SQL File 42* SQL File 43* SQL File 44* SQL File 45* x SQL File 46*

Limit to 1000 rows

```

1 • create view View1 as
2 | select s.student_ID, s.acceptance, s.GRE_Q, s.GRE_V,s.GRE_AWA,s.TOEFL, s.Undergrad_University, s.Undergrad_Major, u.university

```

Result Grid

	student_ID	acceptance	GRE_Q	GRE_V	GRE_AWA	TOEFL	Undergrad_University	Undergrad_Major	university_name	avg_pay
▶	1	Admit	157	142	3.5	97	University of Mumbai	Computer Engineering	Northeastern University	100000
	2	Admit	None	None	None	None	None	0	Northeastern University	100000
	3	Admit	770	460	3	105	fr.c.r.c.e.(bandra)	production engineering	Northeastern University	100000
	4	Admit	159	152	3	94	None	CS	Northeastern University	100000
	5	Admit	158	144	3	100	Apeejay College of Engineering	ECE	Northeastern University	100000
	6	Admit	162	155	4	None	RNSIT	CS	Northeastern University	100000
	7	Admit	160	154	3.5	108	Sri Bhagawan Mahaveer Jain College of Enginee...	Information Science	Northeastern University	100000
	10	Admit	760	680	4	111	Fr CRIT Mumbai University	Mechanical Engineering	Northeastern University	100000
	11	Admit	740	540	5	113	VTU	CS	Northeastern University	100000
	13	Admit	780	590	3.5	114	Manipal Institue of Technoloov	ECE	Northeastern University	100000

6. Display the list of a specific course offered by all the universities.

Description: This query returns a value of all the universities that provide a particular course

Actor: Student

Precondition: The student must select a course

Steps:

Actor Action: The student selects a particular course that he wishes to be enrolled for

System Response: The system displays a list of all the universities that provide that particular course

Postcondition: A list of all the universities that offer analytics courses is displayed

SQL Query:

create view view2 as

select c.course_name, u.university_name from course c inner join

university_details u on c.University_ID = u.University_ID where c.course_name

like '%Analytics%';

The screenshot shows a SQL IDE with multiple tabs. The active tab displays a SQL query to create a view named 'view2'. The query selects course names and university names from a 'course' table, joined with 'university_details' on 'University_ID', and filters for course names containing 'Analytics'. Below the query editor, the 'Result Grid' shows the output of the query.

course_name	university_name
Analytics	Northeastern University
Applied Analytics	Northeastern University
graduate-certificate-in-urban-analytics-14299	Northeastern University
Masters of Data Analytics	Northeastern University
Management - Business Analytics MS	University of Alberta
Biomedical Data Analytics	Arizona State University
Analytics	Arizona State University

7. Display the data about previous admits where the year of joining \geq 2015

Description: To view a list of all the previous admits to a university after 2015

Actor: The student

Precondition: The student must select a year

Steps:

Actor Action: The student selects the year and displays the admits

System Response: The system displays a list of all the previous admits

Postcondition: The list is displayed with all the previous admits

SQL Query:

```
create view view5 as
select s.student_id, ud.university_name as
Target_University,s.Target_Major,s.Year_of_Joining,s.GRE_Total,s.TOEFL,u.G
RE as University_GRE_Cutoff,U.TOEFL as
University_TOEFL_cutoff,s.acceptance
from student s, university u, university_details ud
where u.University_ID = s.University_ID and ud.University_ID = u.University_ID
and s.Year_of_Joining>2015;
```

```

1 • create view view5 as
2 select s.student_id, ud.university_name as Target_University,s.Target_Major,s.Year_of_Joining,s.GRE_Total,s.TOEFL,u.GRE as University_G
3 from student s, university u, university_details ud
4 where u.University_ID = s.University_ID and ud.University_ID = u.University_ID and s.Year_of_Joining>2015;
5
6

```

student_id	Target_University	Target_Major	Year_of_Joining	GRE_Total	TOEFL	University_GRE_Cutoff	University_TOEFL_cutoff	acceptance
22	Northeastern University	Computer Science	2016	323	116	310	100	Admit
147	Northeastern University	MIS	2016	307	91	310	100	Admit
255	Northeastern University	MIS	2016	306	100	310	100	Admit
377	Northeastern University	Industrial Engineering	2016	321	111	310	100	Admit
405	Northeastern University	Computer Science	2016	321	101	310	100	Admit
446	Northeastern University	Computer Science	2016	315	98	310	100	Admit
585	Northeastern University	Engineering Management	2016	319	108	310	100	Admit
654	Northeastern University	Computer Science	2016	314	110	310	100	Admit

8. Display the list of courses choices based on specifications a student can select from post getting the admit.

Description: To view a list of all the possible choices student has

Actor: Student

Precondition: The student must select a course name for acceptance

Steps:

Actor Action: The student must select a course name

System Response: A list is displayed with all the possible admits a student has from a university

Postcondition: A list is displayed with all possible universities

SQL Query:

create view view6 as

select s.student_id, ud.university_name,s.Acceptance,c.Course_Name from
student s join course c on c.University_ID = s.University_ID join university_details
ud on ud.university_id = c.University_ID and s.Acceptance = 'Admit' and
c.Course_Name like '%Engineering%';

The screenshot shows a SQL IDE with multiple tabs. The active tab, 'SQL File 45', contains the following SQL query:

```

1 • create view view6 as
2 select s.student_id, ud.university_name,s.Acceptance,c.Course_Name
3 from student s
4 join course c on c.University_ID = s.University_ID
5 join university_details ud on ud.university_id = c.University_ID
6 and s.Acceptance = 'Admit' and c.Course_Name like '%Engineering%';
7
8

```

Below the query editor, the 'Result Grid' displays the results of the query. The table has four columns: student_id, university_name, Acceptance, and Course_Name. There are 10 rows of data, all with 'Admit' status and 'Engineering' related course names.

student_id	university_name	Acceptance	Course_Name
4	Northeastern University	Admit	master-of-science-in-wireless-and-network-engineering-19163
4	Northeastern University	Admit	graduate-certificate-in-sustainability-engineering-19153
4	Northeastern University	Admit	graduate-certificate-in-supply-chain-engineering-management-14430
5	Northeastern University	Admit	Masters of Software Engineering Management
5	Northeastern University	Admit	master-of-science-in-software-engineering-systems-arlington-19124
5	Northeastern University	Admit	graduate-certificate-in-software-engineering-systems-arlington-19131
5	Northeastern University	Admit	master-of-science-in-software-engineering-systems-18774
5	Northeastern University	Admit	graduate-certificate-in-software-engineering-systems-14354
5	Northeastern University	Admit	master-of-science-in-wireless-and-network-engineering-19163
5	Northeastern University	Admit	graduate-certificate-in-sustainability-engineering-19153

9. Display the list of desired research opportunities available in a university based on the course selection of a student.

Description: To view the list of all the research in a particular course

Actor: Student

Precondition: The student must select a research domain say Machine Learning

Steps:

Actor Action: The student must select a domain for research

System Response: The system displays a list of all the research in a particular course

Postcondition: A list is displayed with all the research going on in a course

SQL Query:

create view view8 as

```

select r.research_name, c.course_name from research r inner join course c
where r.University_ID = c.University_ID and r.Research_Name like '%Machine
Learning%' and course_name in (select Course_Name from course where
course_name like '%Analytics%');

```

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

```

1 • create view view8 as
2   select r.research_name, c.course_name
3   from research r inner join course c
4   where r.University_ID = c.University_ID and r.Research_Name like '%Machine Learning%'
5   and course_name in (select Course_Name from course where course_name like '%Analytics%');

```

Below the editor is the 'Result Grid' tab, which displays the following data:

research_name	course_name
Research in Programming Languages and Machine Learning	Analytics
Research in Programming Languages and Machine Learning	Applied Analytics
Research in Programming Languages and Machine Learning	graduate-certificate-in-urban-analytics-14299
Research in Programming Languages and Machine Learning	Masters of Data Analytics

10. Display the list of research opportunities available in an university based on the location of the university.

Description: To display a list of research going in a particular location in which the university exists

Actor: Student

Precondition: The student must select a location

Steps:

Actor Action: The student selects a location to look for research

System Response: The system displays a list of universities in that location where research is going on

Postcondition: A list is displayed with the research going in a particular location in which the university exists

SQL Query:

```

create view view10 as
select r.research_name, ud.university_name
from research r, university_details ud, university u
where r.university_id = ud.university_id
and u.Location = "Boston";

```

SQL File 36*

SQL File 37*

SQL File 38*

SQL File 40*

SQL File 41

SQL File 42*

SQL File 43*

SQL File 44*

SQL File 45* x

Limit to 1000 rows

```

1 • create view view10 as
2   select r.research_name, ud.university_name
3   from research r, university_details ud, university u
4   where r.university_id = ud.university_id
5   and u.Location = "Boston";

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

research_name	university_name
Research and Fact-Checking for Book Chapter about Global Adaptation and Managed Retreat	Harvard University
Maryland Center for Undergraduate Research (MCUR)	University of Houston, Clear Lake
Research Experience and Applied Learning (REAL) Portal	University of California, Irvine
Interdisciplinary research takes students beyond the classroom	San Jose State University
▶ Collaborations drive discovery	San Jose State University
Promising approaches to global freshwater challenges	San Jose State University
Building a brain: What does it take to make machines as smart as we are?	San Jose State University
Freshman Research Initiative	University of Southern California
Undergraduate Research Journals	University of Southern California
EUREKA Research Database	University of Southern California