


## Part 1

### CREATE DATABASE statements:


```
CREATE DATABASE doctoral;
```

### CREATE TABLE statements:


```
CREATE TABLE PHDSTUDENT(  
    StudentId VARCHAR(255) NOT NULL,  
    FName VARCHAR(255),  
    LName VARCHAR(255),  
    StSem VARCHAR(255),  
    StYear INT,  
    Supervisor VARCHAR(255),  
    PRIMARY KEY(StudentId)  
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>StudentId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>FName</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
3	<b>LName</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
4	<b>StSem</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
5	<b>StYear</b>	int(11)			Yes	NULL		
6	<b>Supervisor</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		


```
CREATE TABLE GRANTS(  
    GrantId VARCHAR(255),  
    Type VARCHAR(255),  
    GrantTitle VARCHAR(255),  
    Sourc VARCHAR(255),  
    StDate DATE,  
    EndDate DATE,  
    StAmount INT,  
    CurrBalance INT,  
    PRIMARY KEY(GrantId)  
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>GrantId</b> 	varchar(255)	utf8mb4_general_ci		No	<i>None</i>		
2	<b>Type</b>	varchar(255)	utf8mb4_general_ci		Yes	<i>NULL</i>		
3	<b>GrantTitle</b>	varchar(255)	utf8mb4_general_ci		Yes	<i>NULL</i>		
4	<b>Sourc</b>	varchar(255)	utf8mb4_general_ci		Yes	<i>NULL</i>		
5	<b>StDate</b>	date			Yes	<i>NULL</i>		
6	<b>EndDate</b>	date			Yes	<i>NULL</i>		
7	<b>StAmount</b>	int(11)			Yes	<i>NULL</i>		
8	<b>CurrBalance</b>	int(11)			Yes	<i>NULL</i>		


```
CREATE TABLE SCHOLARSHIP(
  ScholarshipID VARCHAR(255),
  Type VARCHAR(255),
  Source VARCHAR(255),
  FundingCountry VARCHAR(255),
  FundingOrganization VARCHAR(255),
  PRIMARY KEY(ScholarshipID)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>ScholarshipID</b> 	varchar(255)	utf8mb4_general_ci		No	<i>None</i>		
2	<b>Type</b>	varchar(255)	utf8mb4_general_ci		Yes	<i>NULL</i>		
3	<b>Source</b>	varchar(255)	utf8mb4_general_ci		Yes	<i>NULL</i>		
4	<b>FundingCountry</b>	varchar(255)	utf8mb4_general_ci		Yes	<i>NULL</i>		
5	<b>FundingOrganization</b>	varchar(255)	utf8mb4_general_ci		Yes	<i>NULL</i>		




```
CREATE TABLE INSTRUCTOR(
  InstructorId VARCHAR(255),
  FName VARCHAR(255),
  LName VARCHAR(255),
  StartDate DATE,
  Degree VARCHAR(255),
  Rank VARCHAR(255),
  Type VARCHAR(255),
  PRIMARY KEY(InstructorId)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>InstructorId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>FName</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
3	<b>LName</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
4	<b>StartDate</b>	date			Yes	NULL		
5	<b>Degree</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
6	<b>Rank</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
7	<b>Type</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		


```
CREATE TABLE COURSE(
  CourseID VARCHAR(255),
  CName VARCHAR(255),
  PRIMARY KEY(CourseID)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>CourseID</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>CName</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		



```
CREATE TABLE SECTION(
  SectionId VARCHAR(255),
  CourseId VARCHAR(255),
  InstructorId VARCHAR(255),
  TA VARCHAR(255),
  PRIMARY KEY(SectionId, CourseId),
  FOREIGN KEY(CourseId) REFERENCES course(CourseID)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>SectionId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>CourseId</b>  	varchar(255)	utf8mb4_general_ci		No	None		
3	<b>InstructorId</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
4	<b>TA</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		



```
CREATE TABLE MILESTONE(
  MID VARCHAR(255),
  MName VARCHAR(255),
  PRIMARY KEY(MID)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>MID</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>MName</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		

```
CREATE TABLE GRA(
  StudentId VARCHAR(255),
  GrantId VARCHAR(255),
  MonthlyPay INT,
  MajaorResearchArea VARCHAR(255),
  PRIMARY KEY(StudentId),
  FOREIGN KEY(StudentId) REFERENCES phdstudent(StudentId),
  FOREIGN KEY(GrantId) REFERENCES grants(GrantId)
);
```



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>StudentId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>GrantId</b> 	varchar(255)	utf8mb4_general_ci		Yes	NULL		
3	<b>MonthlyPay</b>	int(11)			Yes	NULL		
4	<b>MajaorResearchArea</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		

```
CREATE TABLE GTA(
  SectionID VARCHAR(255),
  MonthlyPay INT,
  Department VARCHAR(255),
  StudentId VARCHAR(255),
  PRIMARY KEY (StudentId),
  FOREIGN KEY (SectionID) REFERENCES section(SectionId),
  FOREIGN KEY (StudentId) REFERENCES phdstudent(StudentId)
);
```


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>SectionID</b> 	varchar(255)	utf8mb4_general_ci		Yes	NULL		
2	<b>MonthlyPay</b>	int(11)			Yes	NULL		
3	<b>Department</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		
4	<b>StudentId</b> 	varchar(255)	utf8mb4_general_ci		No	None		

```
CREATE TABLE SCHOLARSHIPSUPPORT (
  StudentId varchar(255),
  ScholarshipID varchar(255),
  MonthlyStipned float,
  Category varchar(255),
  PRIMARY KEY (StudentId),
```


FOREIGN KEY (ScholarshipID) REFERENCES scholarship (ScholarshipID),  
FOREIGN KEY (StudentId) REFERENCES phdstudent (StudentId)  
);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>StudentId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>ScholarshipID</b> 	varchar(255)	utf8mb4_general_ci		Yes	NULL		
3	<b>MonthlyStipned</b>	float			Yes	NULL		
4	<b>Category</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		



CREATE TABLE SELFSUPPORT (  
StudentId varchar(255),  
TutionPaymentPlan varchar(255),  
PRIMARY KEY (StudentId),  
FOREIGN KEY (StudentId) REFERENCES phdstudent (StudentId)  
);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>StudentId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>TutionPaymentPlan</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL		




CREATE TABLE TT (  
InstructorId varchar(255),  
NoOfPhDStudents int(11),  
PRIMARY KEY (InstructorId),  
FOREIGN KEY (InstructorId) REFERENCES instructor (InstructorId)  
);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>InstructorId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>NoOfPhDStudents</b>	int(11)			Yes	NULL		




CREATE TABLE MILESTONESPASSED (  
StudentId varchar(255) ,  
MId varchar(255) ,  
PassDate date ,  
PRIMARY KEY (StudentId,MId),  
FOREIGN KEY (StudentId) REFERENCES phdstudent (StudentId)  
);

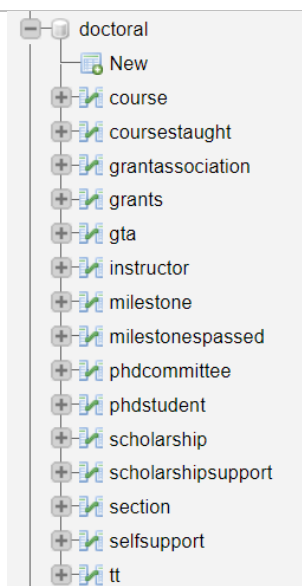
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>StudentId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>MId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
3	<b>PassDate</b>	date			Yes	NULL		

```
CREATE TABLE PHDCOMMITTEE (
  StudentId varchar(255) ,
  InstructorId varchar(255) ,
  PRIMARY KEY (StudentId,InstructorId),
  FOREIGN KEY (StudentId) REFERENCES phdstudent (StudentId),
  FOREIGN KEY (InstructorId) REFERENCES instructor (InstructorId)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>StudentId</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>InstructorId</b>  	varchar(255)	utf8mb4_general_ci		No	None		

```
CREATE TABLE COURSESTAUGHT (
  CourseID varchar(255) ,
  InstructorId varchar(255) ,
  PRIMARY KEY (CourseID,InstructorId),
  FOREIGN KEY (CourseID) REFERENCES course (CourseID),
  FOREIGN KEY (InstructorId) REFERENCES instructor (InstructorId)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>CourseID</b> 	varchar(255)	utf8mb4_general_ci		No	None		
2	<b>InstructorId</b>  	varchar(255)	utf8mb4_general_ci		No	None		



# SELECT statements:

1 SELECT \* FROM PHDSTUDENT;

✓ Showing rows 0 - 24 (73 total, Query took 0.0011 seconds.)

SELECT \* FROM PHDSTUDENT;

StudentId	FName	LName	StSem	StYear	Supervisor
AA1234	Abimbola	Abioye	Fall	2016	AO5671
AA2345	Afia	Achebe	Fall	2016	AO5671
AB2903	Aditya	Bhatt	Fall	2016	AS2348
AD1035	Ashish	Dwivedi	Fall	2017	RB1897
AJ1836	Alice	Jones	Fall	2016	AS2348
AJ1934	Adriana	Johnson	Fall	2016	AO5671
AK1958	Abdul	Kalam	Fall	2018	BL9856
AL8345	Alejandro	Lopez	Fall	2016	AO5671
AM2950	Armand	Martin	Fall	2016	AO5671
AM8051	Antonio	Martinez	Fall	2016	AS2348
AP2578	Anastasia	Petrov	Fall	2017	SB2561
AW1023	Abdurrahman	Wahid	Fall	2017	BL9856
AY2653	Adarsh	Yadav	Fall	2017	AS2348
BD5678	Berko	Dogo	Fall	2017	AO5671
BE1256	Bosede	Eesuola	Fall	2017	AS2348
BM1745	Ben	Mueller	Fall	2016	RB1897
CC6716	Chayton	Chavis	Fall	2018	BL9856
DB1783	Dakota	Benally	Fall	2016	AO5671
DB6291	Dyani	Begay	Fall	2016	AO5671
EN4589	Emeka	Nenge	Fall	2018	RB1897
FE6789	Fumnaya	Egebe	Fall	2017	AS2348
FK8907	Folami	Kanye	Fall	2017	RB1897
FS6892	Francisco	Sanchez	Fall	2018	AO5671
GD1948	Gabriel	Durand	Fall	2017	RB1897
GM6294	Gabriella	Miller	Fall	2017	RB1897

2 SELECT \* FROM GRANTS;


✓ Showing rows 0 - 3 (4 total, Query took 0.0010 seconds.)

SELECT \* FROM GRANTS;

GrantId	Type	GrantTitle	Sourc	StDate	EndDate	StAmount	CurrBalance
DOE1	Educational	GAANN	US Department of Education	2015-01-01	2022-08-30	1500000	500000
DOT4	Research	University Transportation	U.S. Department of Transportation	2016-01-01	2021-12-31	1500000	400000
NIH3	Research	Resistance to Infections	National Institutes of Health	2018-09-01	2024-08-30	450000	375000
NSF2	Research	Undergraduate Experiences	National Science Foundation	2018-09-01	2021-08-30	300000	250000

3

SELECT \* FROM SCHOLARSHIP;


 Showing rows 0 - 9 (10 total, Query took 0.0011 seconds.)

SELECT \* FROM SCHOLARSHIP;

ScholarshipID	Type	Source	FundingCountry	FundingOrganization
SCH1001	Merit-Based	Government	USA	National Science Foundation
SCH1002	Need-Based	Private	Canada	The Gates Scholarship
SCH1003	Research	University	UK	Cambridge Trust
SCH1004	Leadership	Non-Profit	Australia	Rotary Foundation
SCH1005	Talent	Corporate	Germany	Siemens AG
SCH1006	International	Government	Japan	MEXT
SCH1007	Diversity	Private	Sweden	Spotify Inclusion Fund
SCH1008	Athletic	University	USA	NCAA
SCH1009	Artistic	Non-Profit	France	Louvre Endowment
SCH1010	STEM	Corporate	South Korea	Samsung Innovation Fund

4

SELECT \* FROM INSTRUCTOR;

 Showing rows 0 - 11 (12 total, Query took 0.0014 seconds.)

SELECT \* FROM INSTRUCTOR;

InstructorId	FName	LName	StartDate	Degree	Rank	Type
AD4267	Andrea	Delgado	2002-07-15	PhD	Associate Professor	Adjunct
AO1290	Asu	Ozdaglar	2002-07-15	PhD	Associate Professor	Adjunct
AO5671	Adegoke	Olubummo	1982-07-19	PhD	Professor	TT
AS2348	Ahmed	Sarhan	1989-07-19	PhD	Professor	TT
BL9856	Barbara	Liskov	2001-07-19	PhD	Professor	TT
CA2876	Corey	Ashley	2002-07-15	Masters	Associate Professor	Adjunct
CC8908	Carlos	Castillo	2017-07-15	PhD	Associate Professor	Adjunct
DP6712	David	Patterson	2001-07-19	PhD	Professor	NTT
HY1945	Henry	Yuen	2008-07-25	PhD	Associate Professor	NTT
RB1897	Ravindran	Balaraman	1990-02-10	PhD	Professor	TT
SB2561	Sanghamitra	Bandyopadhyay	1999-09-15	PhD	Assistant Professor	TT
XZ3456	Xia	Zhou	2005-07-20	PhD	Associate Professor	NTT

5

SELECT \* FROM COURSE;



✓ Showing rows 0 - 12 (13 total, Query took 0.0009 seconds.)

SELECT \* FROM COURSE;

CourseID	CName
CSE1310	Introduction to programming
CSE1320	Intermediate Programming
CSE1325	Object-Oriented Programming
CSE3302	Programming Languages
CSE3310	Fundamentals of Software Engineering
CSE3330	Database 1
CSE3442	Embedded Systems
CSE4351	Parallel Processing
CSE4354	Operating System
CSE5322	Software Design
CSE5324	Artificial Intelligence
CSE6338	Neural Networks
CSE6365	Computer Graphics

6 SELECT \* FROM SECTION;

✓ Showing rows 0 - 24 (33 total, Query took 0.0041 seconds.)

SELECT \* FROM SECTION;

SectionId	CourseId	InstructorId	TA
CSE1310-001	CSE1310	AD4267	TC1038
CSE1320-001	CSE1320	HY1945	PH2091
CSE1325-001	CSE1325	XZ3456	MG1940
CSE3302-001	CSE3302	XZ3456	AD1035
CSE3302-002	CSE3302	HY1945	MP3478
CSE3302-003	CSE3302	AD4267	ML3902
CSE3310-001	CSE3310	CA2876	NN1629
CSE3310-002	CSE3310	AO1290	OT8190
CSE3310-003	CSE3310	CC8908	RS1903
CSE3330-001	CSE3330	AO5671	PG2469
CSE3330-002	CSE3330	AS2348	LM1405
CSE3330-003	CSE3330	RB1897	MA1234
CSE3442-001	CSE3442	SB2561	FS6892
CSE3442-002	CSE3442	BL9856	AK1958
CSE3442-003	CSE3442	DP6712	SJ1947
CSE4351-001	CSE4351	SB2561	SP1938
CSE4351-002	CSE4351	BL9856	AY2653
CSE4351-003	CSE4351	DP6712	SD1946
CSE4354-001	CSE4354	AO5671	JF1092
CSE4354-002	CSE4354	AS2348	PS1904
CSE4354-003	CSE4354	RB1897	SS1304
CSE5322-001	CSE5322	AO5671	JG1036
CSE5322-002	CSE5322	AS2348	VA1938
CSE5322-003	CSE5322	RB1897	AB2903
CSE5324-001	CSE5324	SB2561	JR1930

7 SELECT \* FROM MILESTONE;

✓ Showing rows 0 - 4 (5 total, Query took 0.0009 seconds.)

SELECT \* FROM MILESTONE;

MID	MName
CE	Comprehensive Exam
CM	Committee Formed
DE	Diagnostics Evaluation
DF	Defense
PR	Proposal

8 SELECT \* FROM GRA;

✓ Showing rows 0 - 10 (11 total, Query took 0.0010 seconds.)

SELECT \* FROM GRA;

StudentId	GrantId	MonthlyPay	MajaorResearchArea
AW1023	NIH3	3000	Cyber
CC6716	DOE1	1000	AI
JP9384	DOT4	3000	SE
KH1029	NIH3	3000	AI
KJ1928	DOT4	1000	Cyber
LY1049	NSF2	2000	Cyber
RK1489	NSF2	3000	SE
RY1726	DOT4	2000	AI
WW3847	DOE1	3000	AI
XY2938	DOE1	3000	SE
ZW1029	DOE1	3000	Cyber

9 SELECT \* FROM GTA;

✓ Showing rows 0 - 24 (33 total, Query took 0.0037 seconds.)

SELECT \* FROM GTA;

SectionID	MonthlyPay	StudentId	Department
CSE5322-003	2000	AB2903	CSE
CSE3302-001	1500	AD1035	CSE
CSE3442-002	1000	AK1958	CSE
CSE6338-001	2000	AL8345	CSE
CSE6338-002	2000	AM8051	CSE
CSE4351-002	1500	AY2653	CSE
CSE6365-003	2000	DB1783	CSE
CSE6365-002	2000	DB6291	CSE
CSE3442-001	1000	FS6892	CSE
CSE4354-001	1500	JF1092	CSE
CSE5322-001	2000	JG1036	CSE
CSE5324-003	2000	JG8927	CSE
CSE5324-001	2000	JR1930	CSE
CSE6338-003	2000	KY4892	CSE
CSE3330-002	1000	LM1405	CSE
CSE3330-003	1000	MA1234	CSE
CSE5324-002	2000	MA1968	CSE
CSE1325-001	1000	MG1940	CSE
CSE3302-003	1500	ML3902	CSE
CSE6365-001	2000	MN2903	CSE
CSE3302-002	1500	MP3478	CSE
CSE3310-001	1500	NN1629	CSE
CSE3310-002	1500	OT8190	CSE
CSE3330-001	1500	PG2469	CSE
CSE1320-001	1000	PH2091	CSE

10 SELECT \* FROM SCHOLARSHIPSUPPORT;

✓ Showing rows 0 - 15 (16 total, Query took 0.0011 seconds.)

SELECT \* FROM SCHOLARSHIPSUPPORT;

StudentId	ScholarshipID	MonthlyStipend	Category
AJ1836	SCH1001	1000	local
AM2950	SCH1003	1200	international
AP2578	SCH1002	800	international
BM1745	SCH1003	1200	international
GD1948	SCH1001	1000	local
GM6294	SCH1001	1000	local
HA2680	SCH1002	800	international
IS2945	SCH1002	800	international
JT6936	SCH1003	1200	international
LB2673	SCH1003	1200	international
LF150	SCH1003	1200	international
MI1589	SCH1002	800	international
NS4289	SCH1003	1200	international
RW9267	SCH1001	1000	local
SC1894	SCH1002	800	international
VB1059	SCH1001	1000	local

11 SELECT \* FROM SELFSUPPORT;

✓ Showing rows 0 - 12 (13 total, Query took 0.0014 seconds.)

SELECT \* FROM SELFSUPPORT;

StudentId	TuitionPaymentPlan
AA1234	Installments
AA2345	Full Payment
AJ1934	Full Payment
BD5678	Installments
BE1256	Full Payment
EN4589	Full Payment
FE6789	Installments
FK8907	Installments
IJ1569	Full Payment
KO1905	Installments
MS1259	Installments
NZ1278	Full Payment
VW8930	Installments

12 SELECT \* FROM TT;

✓ Showing rows 0 - 4 (5 total, Query took 0.0011 seconds.)

SELECT \* FROM TT;

InstructorId	NoOfPhDStudents
AO5671	10
AS2348	20
BL9856	9
RB1897	13
SB2561	21

13 SELECT \* FROM MILESTONESPASSED;

✓ Showing rows 0 - 24 (61 total, Query took 0.0011 seconds.)

SELECT \* FROM MILESTONESPASSED;

StudentId	MId	PassDate
AA1234	CM	2016-12-01
AA2345	CM	2016-12-01
AB2903	CM	2016-12-05
AJ1836	CM	2016-12-02
AJ1934	CM	2016-12-02
AK1958	CM	2017-12-07
AL8345	CM	2016-12-05
AM2950	CM	2016-12-05
AM8051	CM	2016-12-06
BD5678	CM	2017-12-07
BE1256	CM	2017-12-07
BM1745	CM	2016-12-05
DB1783	CM	2016-12-07
DB1783	DE	2017-06-05
DB6291	CM	2016-12-07
DB6291	DE	2017-06-06
EN4589	CM	2018-12-06
FE6789	CM	2017-12-07
FK8907	CE	2018-05-07
FK8907	CM	2017-12-07
FK8907	DE	2017-12-07
HA2680	CM	2018-12-08
IJ1569	CM	2017-12-07
IJ1569	DE	2017-12-06
IS2945	CM	2016-12-02

14 SELECT \* FROM PHDCOMMITTEE;

✓ Showing rows 0 - 24 (80 total, Query took 0.0009 seconds.)

SELECT \* FROM PHDCOMMITTEE;

StudentId	InstructorId
AA1234	AS2348
AA1234	RB1897
AA2345	BL9856
AA2345	SB2561
AB2903	RB1897
AB2903	SB2561
AJ1836	BL9856
AJ1836	SB2561
AJ1934	AS2348
AJ1934	RB1897
AK1958	AO5671
AK1958	AS2348
AL8345	AS2348
AL8345	RB1897
AM2950	BL9856
AM2950	SB2561
AM8051	BL9856
AM8051	SB2561
BD5678	RB1897
BD5678	SB2561
BE1256	AO5671
BE1256	RB1897
BE1256	SB2561
BM1745	AS2348
BM1745	BL9856

15 SELECT \* FROM COURSESTAUGHT;

✓ Showing rows 0 - 24 (25 total, Query took 0.0051 seconds.)

SELECT \* FROM COURSESTAUGHT;

CourseID	InstructorId
CSE1320	AO1290
CSE1325	AO1290
CSE1325	CA2876
CSE3302	XZ3456
CSE3310	HY1945
CSE3330	AD4267
CSE3442	AD4267
CSE3442	CA2876
CSE4351	DP6712
CSE4351	SB2561
CSE4351	XZ3456
CSE4354	DP6712
CSE4354	HY1945
CSE4354	SB2561
CSE5322	AS2348
CSE5322	BL9856
CSE5324	AO5671
CSE5324	AS2348
CSE5324	RB1897
CSE6338	AO5671
CSE6338	AS2348
CSE6338	RB1897
CSE6365	BL9856
CSE6365	RB1897
CSE6365	SB2561


## Part 2

### SQL Statements:

1

**Retrieve the full names of all instructors who have a PhD degree.**

**Query:** SELECT CONCAT(FName,' ',LName) as FullName FROM instructor WHERE Degree='PhD';

 Showing rows 0 - 10 (11 total, Query took 0.0030 seconds.)

```
SELECT CONCAT(FName, ' ',LName) as FullName FROM instructor WHERE Degree='PhD';
```

FullName
Andrea Delgado
Asu Ozdaglar
Adegoke Olubummo
Ahmed Sarhan
Barbara Liskov
Carlos Castillo
David Patterson
Henry Yuen
Ravindran Balaraman
Sanghamitra Bandyopadhyay
Xia Zhou

2

**Find the names of all PhD students who are also working as Graduate Teaching Assistants (GTAs).**

**Query:** SELECT CONCAT(FName,' ',LName) AS Name FROM phdstudent WHERE StudentId IN (SELECT StudentId FROM gta);



Showing rows 0 - 24 (33 total, Query took 0.0020 seconds.)

```
SELECT CONCAT(FName, ' ', LName) AS Name FROM phdstudent WHERE StudentId IN (SELECT StudentId FROM gta);
```

Name
Tala Castillo
Pocahontas Hill
Maanas Gandhi
Ashish Dwivedi
Maria Perez
Malia Locklear
Nokomis Nez
Odina Toledo
Rosa Sanchez
Patricia Gomez
Lucinda Moreau
Miguel Angel
Francisco Sanchez
Abdul Kalam
Salil Jha
Sai Paranjape
Adarsh Yadav
Shalini Desai
Josefina Flores
Parvinder Singh
Simarjeet Suri
Juan Gonzalez
Vivaan Ahuja
Aditya Bhatt
Jesus Rodriguez

3

Retrieve a list of all courses taught by a specific instructor with InstructorID = BL9856.  
Query: SELECT CName FROM course WHERE CourseID IN (SELECT CourseID FROM coursestaught WHERE InstructorId='BL9856');

Showing rows 0 - 1 (2 total, Query took 0.0028 seconds.)

```
SELECT CName FROM course WHERE CourseID IN (SELECT CourseID FROM coursestaught WHERE InstructorId='BL9856');
```

CName
Software Design
Computer Graphics

4

Show the total number of PhD students supervised by each TT instructor. If a TT instructor has not supervised any students so far, display a zero for them.  
Query: SELECT instructor.InstructorId, ifnull(tt.NoOfPhDStudents,0) as NoOfPhDStudents FROM ( instructor LEFT OUTER JOIN tt ON instructor.InstructorId = tt.InstructorId );

Showing rows 0 - 11 (12 total, Query took 0.0040 seconds.)

```
SELECT instructor.InstructorId, ifnull(tt.NoOfPhDStudents,0) as NoOfPhDStudents FROM ( instructor LEFT OUTER JOIN tt ON instructor.InstructorId = tt.InstructorId );
```

InstructorId	NoOfPhDStudents
AD4267	0
AO1290	0
AO5671	10
AS2348	20
BL9856	9
CA2876	0
CC8908	0
DP6712	0
HY1945	0
RB1897	13
SB2561	21
XZ3456	0

5

Showing rows 0 - 24 (38 total, Query took 0.0036 seconds.)

```
SELECT * FROM phdstudent WHERE phdstudent.StudentId in (select milestonespassed.StudentId from milestonespassed WHERE milestonespassed.Mid='CM');
```



StudentId	FName	LName	StSem	StYear	Supervisor
AA1234	Abimbola	Abioye	Fall	2016	AO5671
AA2345	Afia	Achebe	Fall	2016	AO5671
AB2903	Aditya	Bhatt	Fall	2016	AS2348
AJ1836	Alice	Jones	Fall	2016	AS2348
AJ1934	Adriana	Johnson	Fall	2016	AO5671
AK1958	Abdul	Kalam	Fall	2018	BL9856
AL8345	Alejandro	Lopez	Fall	2016	AO5671
AM2950	Armand	Martin	Fall	2016	AO5671
AM8051	Antonio	Martinez	Fall	2016	AS2348
BD5678	Berko	Dogo	Fall	2017	AO5671
BE1256	Bosede	Eesuola	Fall	2017	AS2348
BM1745	Ben	Mueller	Fall	2016	RB1897
DB1783	Dakota	Benally	Fall	2016	AO5671
DB6291	Dyani	Begay	Fall	2016	AO5671
EN4589	Emeka	Nenge	Fall	2018	RB1897
FE6789	Fumnaya	Egebe	Fall	2017	AS2348
FK8907	Folami	Kanye	Fall	2017	RB1897
HA2680	Hector	Adams	Fall	2018	SB2561
IJ1569	Ife	Jelani	Fall	2017	RB1897
IS2945	Ivan	Sokolov	Fall	2016	AO5671
JG1036	Juan	Gonzalez	Fall	2016	BL9856
JG8927	Jose	Garcia	Fail	2016	AO5671
JP9384	Jung	Park	Fall	2016	AO5671
JR1930	Jesus	Rodriguez	Fall	2016	AS2348
KH1029	Ki	Hajar	Fall	2016	BL9856

Sakshi Patel (saks)

6

Retrieve a list of details such as GrantId, GrantTitle, Source, StDate, EndDate, StAmount, and CurrBalance for all grants that started in the year 2018.

Query: select \* from GRANTS where YEAR(grants.StDate)=2018;

7	<div>  Showing rows 0 - 1 (2 total, Query took 0.0035 seconds.)         </div> <div> <pre>select * from GRANTS where YEAR(grants.StDate)=2018;</pre> </div>							
	GrantId	Type	GrantTitle	Source	StDate	EndDate	StAmount	CurrBalance
	NIH3	Research	Resistance to Infections	National Institutes of Health	2018-09-01	2024-08-30	450000	375000
8	NSF2	Research	Undergraduate Experiences	National Science Foundation	2018-09-01	2021-08-30	300000	250000
	<div>  Showing rows 0 - 12 (13 total, Query took 0.0027 seconds.)         </div> <div> <pre>select * from phdstudent WHERE phdstudent.StudentId in (select selfsupport.StudentId from selfsupport);</pre> </div>							
	StudentId	FName	LName	StSem	StYear	Supervisor		
	AA1234	Abimbola	Abioye	Fall	2016	AO5671		
	AA2345	Afia	Achebe	Fall	2016	AO5671		
	AJ1934	Adriana	Johnson	Fall	2016	AO5671		
	BD5678	Berko	Dogo	Fall	2017	AO5671		
	BE1256	Bosede	Eesuola	Fall	2017	AS2348		
	EN4589	Emeka	Nenge	Fall	2018	RB1897		
	FE6789	Fumnaya	Egebe	Fall	2017	AS2348		
	FK8907	Folami	Kanye	Fall	2017	RB1897		
	IJ1569	Ife	Jelani	Fall	2017	RB1897		
	KO1905	Kojo	Okoro	Fall	2018	SB2561		
	MS1259	Maya	Smith	Fall	2016	AO5671		
	NZ1278	Nia	Zivai	Fall	2018	SB2561		
	VW8930	Violet	Williams	Fall	2016	AO5671		

Showing rows 0 - 15 (16 total, Query took 0.0012 seconds.)

```
SELECT StudentId, MonthlyStipned FROM scholarshipssupport;
```

StudentId	Monthly Stipned
AJ1836	1000
AM2950	1200
AP2578	800
BM1745	1200
GD1948	1000
GM6294	1000
HA2680	800
IS2945	800
JT6936	1200
LB2673	1200
LF150	1200
MI1589	800
NS4289	1200
RW9267	1000
SC1894	800
VB1059	1000

9

**Identify all instructors who are not supervising any PhD students.**  
**SQL:** select Instructor.InstructorId from instructor where instructor.InstructorId not in (SELECT DISTINCT(phdcommittee.InstructorId) from phdcommittee);

Showing rows 0 - 6 (7 total, Query took 0.0043 seconds.)

```
select Instructor.InstructorId from instructor where instructor.InstructorId not in (SELECT DISTINCT(phdcommittee.InstructorId) from phdcommittee);
```

InstructorId
AD4267
AO1290
CA2876
CC8908
DP6712
HY1945
XZ3456

10

**Retrieve a list of all courses that have never been taught (i.e., not listed in the COURSESTAUGHT table).**  
**SQL:** select course.CourseID,course.CName from course where course.CourseID not in (select DISTINCT(coursestaught.CourseID) from coursestaught);

Showing rows 0 - 0 (1 total, Query took 0.0031 seconds.)

```
select course.CourseID,course.CName from course where course.CourseID not in (select DISTINCT(coursestaught.CourseID) from coursestaught);
```

CourseID	CName
CSE1310	Introduction to programming

11

**Retrieve a list of the average monthly pay for Graduate Research Assistants in each major research area.**

**SQL:** select avg(gra.MonthlyPay) as AvgMonthlyPay,gra.MajaorResearchArea from gra group by gra.MajaorResearchArea;

Showing rows 0 - 2 (3 total, Query took 0.0027 seconds.)

```
select avg(gra.MonthlyPay) as AvgMonthlyPay,gra.MajaorResearchArea from gra group by gra.MajaorResearchArea;
```

AvgMonthlyPay	MajaorResearchArea
2250.0000	AI
2250.0000	Cyber
3000.0000	SE

12

**Retrieve a list of full names and the number of sections each instructor is teaching.**

**SQL:** select count(section.SectionId) as TotalSection,instructor.FName,instructor.LName from section JOIN instructor on section.InstructorId=instructor.InstructorId GROUP BY section.InstructorId;

Showing rows 0 - 11 (12 total, Query took 0.0031 seconds.)

```
select count(section.SectionId) as TotalSection,instructor.FName,instructor.LName from section JOIN instructor on section.InstructorId=instructor.InstructorId GROUP BY section.InstructorId;
```

TotalSection	FName	LName
3	Andrea	Delgado
2	Asu	Ozdaglar
3	Adegoke	Olubummo
3	Ahmed	Sarhan
3	Barbara	Liskov
2	Corey	Ashley
2	Carlos	Castillo
3	David	Patterson
3	Henry	Yuen
3	Ravindran	Balaraman
3	Sanghamitra	Bandyopadhyay
3	Xia	Zhou

13

**Retrieve a list of all students who have received a scholarship from ' National Science Foundation'.**

**SQL:** select \* from phdstudent where phdstudent.StudentId in (select scholarshipsupport.StudentId from scholarshipsupport where scholarshipsupport.ScholarshipID in (SELECT scholarship.ScholarshipID from scholarship where scholarship.FundingOrganization like '%National Science Foundation%'));

Showing rows 0 - 4 (5 total, Query took 0.0047 seconds.)

```
select * from phdstudent where phdstudent.StudentId in (select scholarshipsupport.StudentId from scholarshipsupport where scholarshipsupport.ScholarshipID in (SELECT scholarship.ScholarshipID from scholarship where scholarship.FundingOrganization like '%National Science Foundation%'));
```

StudentId	FName	LName	StSem	StYear	Supervisor
AJ1836	Alice	Jones	Fall	2016	AS2348
GD1948	Gabriel	Durand	Fall	2017	RB1897
GM6294	Gabriella	Miller	Fall	2017	RB1897
RW9267	Raymond	Wilson	Fall	2017	AS2348
VB1059	Victor	Brown	Fall	2017	AS2348

14

**Execute a command to delete a record that violates a referential integrity constraint. State the message produced by the DBMS.**

**SQL:** DELETE FROM phdstudent WHERE phdstudent.StudentId='AA1234';

**Message:** Cannot delete or update a parent row: a foreign key constraint fails (`doctoral`.`milestonespassed`, CONSTRAINT `milestonespassed\_ibfk\_1` FOREIGN KEY (`StudentId`) REFERENCES `phdstudent` (`StudentId`))

Error

SQL query: [Copy](#)

```
DELETE FROM phdstudent WHERE phdstudent.StudentId='AA1234';
```

MySQL said:

```
#1451 - Cannot delete or update a parent row: a foreign key constraint fails (`doctoral`.`milestonespassed`, CONSTRAINT `milestonespassed_ibfk_1` FOREIGN KEY (`StudentId`) REFERENCES `phdstudent` (`StudentId`))
```

15

**Execute an update command for Instructor table that attempts to update a record and thereby violates the foreign key constraint.State the message produced by the DBMS.**

**SQL:** UPDATE instructor SET instructor.InstructorId='AA1111' WHERE instructor.InstructorId='AO5671';

**Message:** Cannot delete or update a parent row: a foreign key constraint fails (`doctoral`.`coursestaught`, CONSTRAINT `coursestaught\_ibfk\_2` FOREIGN KEY (`InstructorId`) REFERENCES `instructor` (`InstructorId`))

Error

SQL query: [Copy](#)

```
UPDATE instructor SET instructor.InstructorId='AA1111' WHERE instructor.InstructorId='AO5671';
```

MySQL said:


```
#1451 - Cannot delete or update a parent row: a foreign key constraint fails (`doctoral`.`coursestaught`, CONSTRAINT `coursestaught_ibfk_2` FOREIGN KEY (`InstructorId`) REFERENCES `instructor` (`InstructorId`))
```

- 16 **Execute 3 insert commands for Instructor or TT tables that attempt to insert records, such that the records violate the explicit schema-based constraints (Key, Entity Integrity, Referential Integrity constraints). Make each of the 3 records violate a different types of integrity constraint. Include the insert statements and the error messages produced.**
- a) Violates Key constraint.  
b) Violates Entity Integrity constraint.  
c) Violates Referential Integrity constraint.
- SQL: (a) Violates Key Constraint**  
INSERT INTO phdstudent(StudentId, FName, LName, StSem, StYear, Supervisor) VALUES ('AA2345','Sakshi','Patel','Fall','2017','AO5671');  
**Message:** Duplicate entry 'AA2345' for key 'PRIMARY'

#### Error

SQL query: [Copy](#)

```
INSERT INTO phdstudent(StudentId, FName, LName, StSem, StYear, Supervisor) VALUES ('AA2345','Sakshi','Patel','Fall','2017','AO5671');
```

MySQL said: 

#1062 - Duplicate entry 'AA2345' for key 'PRIMARY'


#### (b) Violates Entity Integrity Constraint

INSERT INTO phdstudent(StudentId, FName, LName, StSem, StYear, Supervisor) VALUES (null,'Sakshi','Patel','Fall','2017','AO5671');  
**Message:** Column 'StudentId' cannot be null

#### Error

SQL query: [Copy](#)

```
INSERT INTO phdstudent(StudentId, FName, LName, StSem, StYear, Supervisor) VALUES (null,'Sakshi','Patel','Fall','2017','AO5671');
```

MySQL said: 

#1048 - Column 'StudentId' cannot be null


#### (c) Violates Referential Integrity Constraint

INSERT INTO scholarshipsupport(StudentId, ScholarshipID, MonthlyStipend, Category) VALUES ('AA1234','SCH1011','1200.00','local');  
**Message:** Cannot add or update a child row: a foreign key constraint fails (`doctoral`.`scholarshipsupport`, CONSTRAINT `scholarshipsupport\_ibfk\_1` FOREIGN KEY (`ScholarshipID`) REFERENCES `scholarship` (`ScholarshipID`))

## Error

SQL query: [Copy](#)

```
INSERT INTO scholarshipsupport(StudentId, ScholarshipID, MonthlyStipned, Category) VALUES ('AA1234','SCH1011','1
```

MySQL said: 

```
#1452 - Cannot add or update a child row: a foreign key constraint fails
(`doctoral`.`scholarshipsupport`, CONSTRAINT `scholarshipsupport_ibfk_1` FOREIGN KEY
(`ScholarshipID`) REFERENCES `scholarship` (`ScholarshipID`))
```