

# Database project

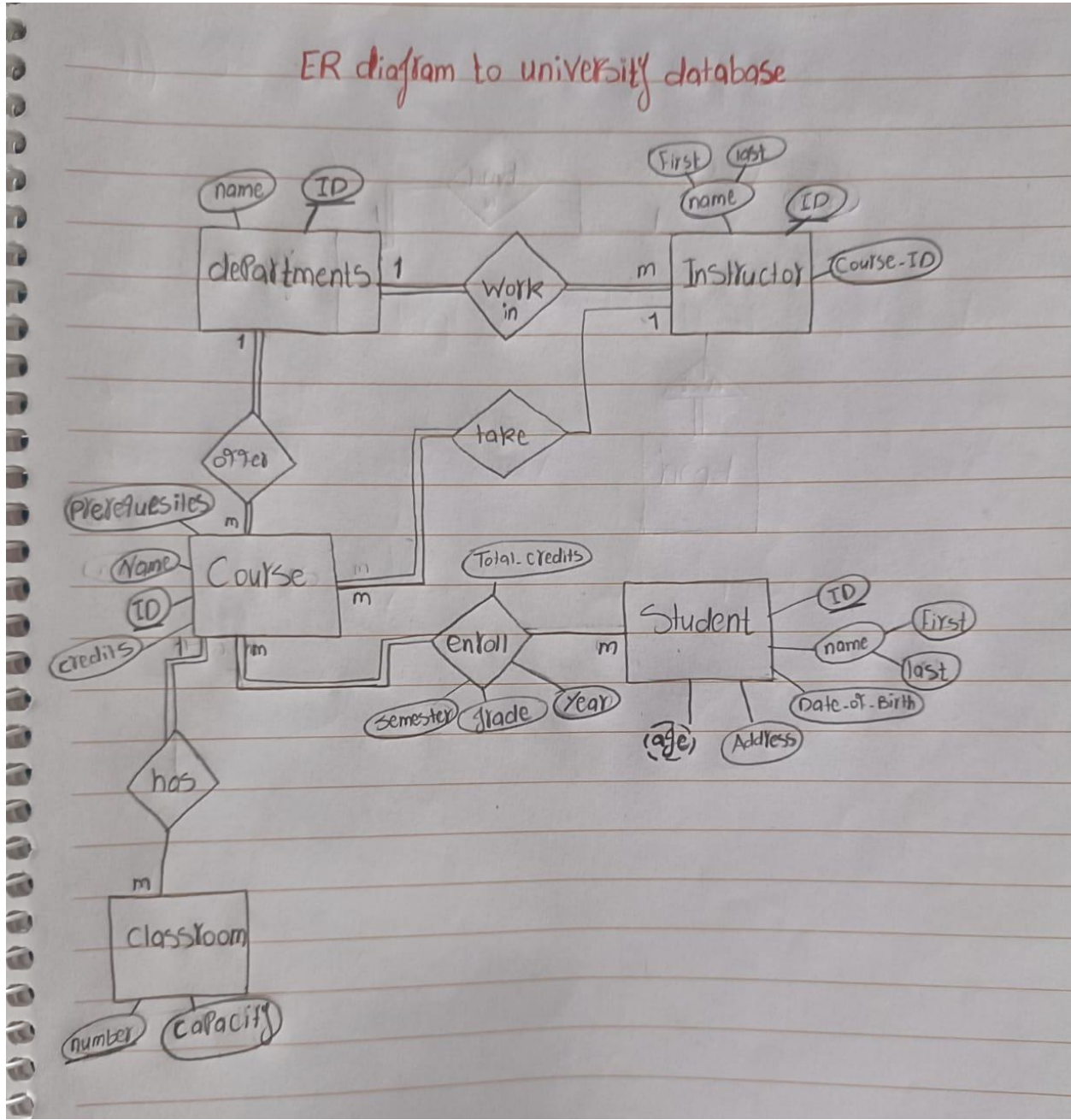
Nureen Ehab Mahmoud Mohamed Barakat

20221465124

Intelligent Systems

# University Database

## ER Diagram:



optional: \_\_\_\_\_

complete: \_\_\_\_\_

## Schema:

**Composite attribute:** will be removed and attributes inside it will be separated to different attributes in our schema

Example: **First\_Name** , **Last\_Name** those are different attributes in schema instead of composite attribute **Name** in ER diagram

**Derived attribute:** will be removed

Example: **age** will be removed

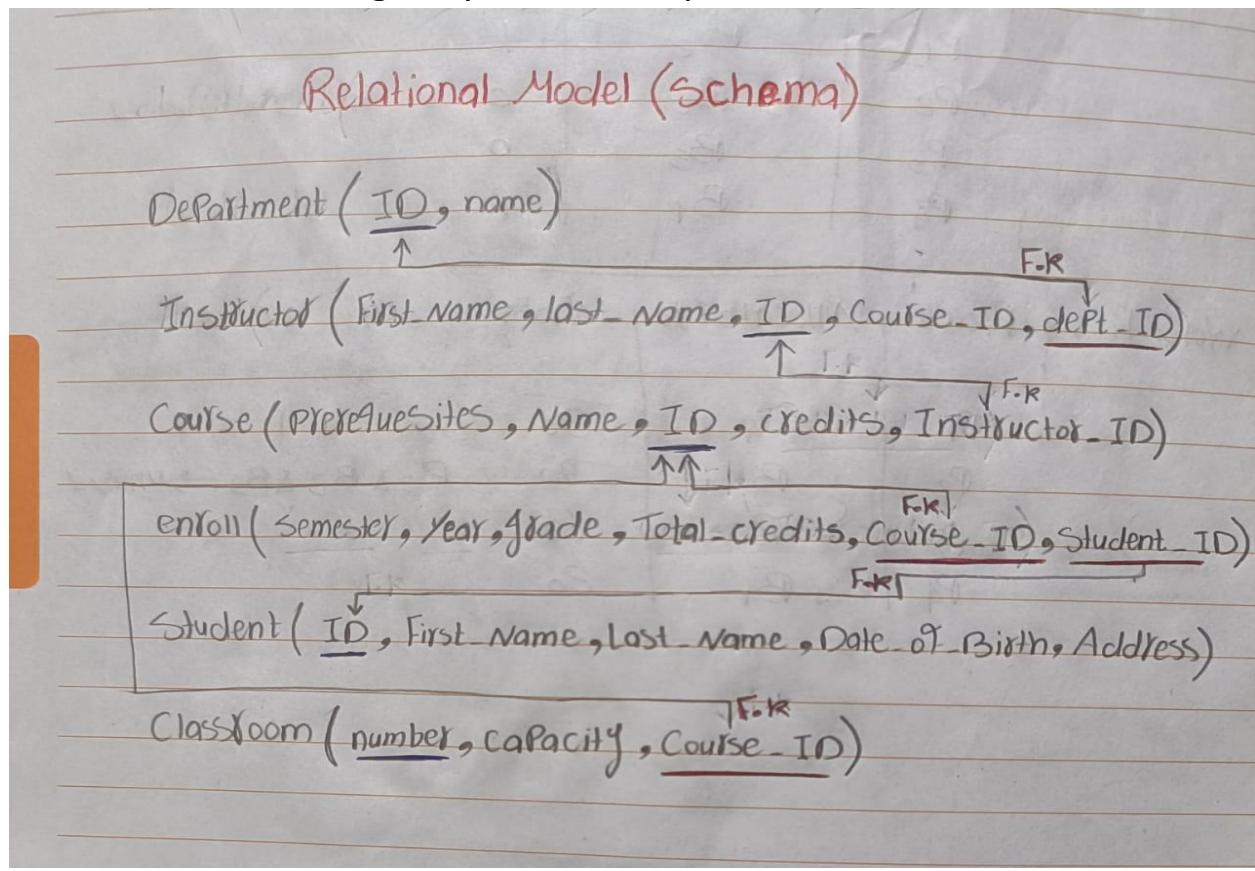
### **Relations:**

#### 1. One to many:

take primary key of **one** relation side to be foreign key in **many** relation side

#### 2. Many to many:

the relation must be different entity and we take primary keys of 2 entities as foreign key in the entity of the relation



## Code:

1) Create university database using **CREATE DATABASE**:

```
CREATE DATABASE university
```

2) Create department table using **CREATE TABLE**

- Inside it we will put the name of each column and its data type
- We will make ID column primary key of department table using **PRIMARY KEY** as each ID is unique

```
CREATE Table department(ID int PRIMARY KEY, Name varchar(30))
```

- Insert values in department table using **INSERT INTO table\_Name** , and it takes name of each column, and then put values inside each column using **VALUES**:

```
INSERT INTO department(ID,Name) VALUES (1,'Intelligent Systems'),  
(2,'cyber security'),  
(3,'business Analytics'),  
(4,'Media Analytics'),  
(5,'Healthcare Analytics')
```

## Output:

ID	Name
1	Intelligent Systems
2	cyber security
3	business Analytics
4	Media Analytics
5	Healthcare Analytics

### 3) Create instructor table using **CREATE TABLE**

- Inside it we will put the name of each column and its data type
- We will make ID column primary key of department table using **PRIMARY KEY** as each ID is unique
- We will put department ID as foreign key using **FOREIGN KEY**(any name of column we want) **REFERENCES** table we take its primary key as foreign key of instructor table (name of primary key column)

Department ID is the foreign key of instructor table as relation between these 2 tables is one to many so we take primary key of **one** relation side (department) to be foreign key in **many** relation side (instructor)

```
CREATE Table instructor(ID int PRIMARY KEY, First_Name varchar(30) , Last_Name varchar(30) , course_ID int , Dept_ID int ,  
                        FOREIGN KEY(Dept_ID) REFERENCES department(ID))
```

- Insert values in instructor table using **INSERT INTO table\_Name**, and it takes name of each column, and then put values inside each column using **VALUES**:

```
INSERT INTO instructor(ID, First_Name, Last_Name , course_ID ,Dept_ID) VALUES(1,'Marwan','Tork',03204,1),  
(2,'Mervat','Mikhail',00101,3),(3,'Eslam','kabani',00105,4),(4,'Yasser','Fouad',00107,1),  
(5,'Nader','Mahmoud',03203,2),(6,'Weam','Zohdy',03202,5),(7,'Amal','Aboulhassan',00104,5),  
(8,'Mohamed','khamis',00106,2),(9,'Mohamed','Waleed',00102,4),(10,'Mohamed','Alhabrouk',00103,3),  
(11,'Shahira','Radwan',00108,2),(12,'Bothina','Mohamed',00203,2),(13,'Mohamed','Kholif',00303,2),  
(14,'Mohamed','Elkholy',00201,2),(15,'Mohamed','Sameh',00109,1),(16,'Ahmed','Haggag',00110,4),  
(17,'Ahmed','younis',00201,2),(18,'Mohamed','Abokhashba',100,5),(19,'Khaled','Elsherif',0001,3),  
(20,'Mohamed','Soliman',0002,1),(21,'Mohamed','Elseraty',0005,3)
```

## Output:

ID	First_Name	Last_Name	course_ID	Dept_ID
1	Marwan	Tork	3204	1
2	Mervat	Mikhail	101	3
3	Eslam	kabani	105	4
4	Yasser	Fouad	107	1
5	Nader	Mahmoud	3203	2
6	Weam	Zohdy	3202	5
7	Amal	Aboulhassan	104	5
8	Mohamed	khamis	106	2
9	Mohamed	Waleed	102	4
10	Mohamed	Alhabrouk	103	3
11	Shahira	Radwan	108	2
12	Bothina	Mohamed	203	2
13	Mohamed	Kholif	303	2
14	Mohamed	Elkholy	201	2
15	Mohamed	Sameh	109	1
16	Ahmed	Haggag	110	4
17	Ahmed	younis	201	2
18	Mohamed	Abokhashba	100	5
19	Khaled	Elsherif	1	3
20	Mohamed	Soliman	2	1
21	Mohamed	Elseraty	5	3

#### 4) Create student table using **CREATE TABLE**

- Inside it we will put the name of each column and its data type
- We will make ID column primary key of department table using **PRIMARY KEY** as each ID is unique

```
CREATE Table student(ID int PRIMARY KEY, First_Name varchar(30) , Last_Name varchar(30) ,  
Date_of_Birth varchar(255) , Address varchar(255))
```

- Insert values in student table using **INSERT INTO table\_Name**, and it takes name of each column, and then put values inside each column using **VALUES**:

```
INSERT INTO student(ID, First_Name, Last_Name , Date_of_Birth , Address)  
VALUES(1465124, 'Nureen', 'Barakat', '7/29/2003', 'Sababasha'), (1465125, 'Yassimen', 'Mohamed', '4/5/2003', 'Smouha'),  
(1465126, 'Noha', 'Mohamed', '6/1/2003', 'Smouha'), (1465143, 'Nadine', 'Tarek', '12/8/2004', 'Sidi Beshr'),  
(1465127, 'Doha', 'Mohamed', '5/23/2003', 'Smouha'), (1465128, 'Reem', 'Mohamed', '2/25/2003', 'Sababasha'),  
(1465129, 'lana', 'Mohamed', '6/25/2003', 'Sababasha'), (1465145, 'Menna', 'Ehab', '6/6/2004', 'Sidi Beshr'),  
(1465130, 'Merna', 'Barakat', '9/3/2002', 'Sababasha'), (1465131, 'Sherif', 'Mohamed', '10/3/2002', 'Smouha'),  
(1465132, 'Hesham', 'Mohamed', '10/2/2002', 'Sababasha'), (1465133, 'Amr', 'Mohamed', '10/23/2002', 'Sababasha'),  
(1465134, 'Mohamed', 'Medhat', '9/23/2002', 'roushdy'), (1465135, 'Ahmed', 'Medhat', '7/25/2002', 'roushdy'),  
(1465136, 'Ahmed', 'Medhat', '8/6/2002', 'Miami'), (1465137, 'Mohamed', 'Ezz', '11/15/2002', 'Sedi Gaber'),  
(1465138, 'Fatma', 'Hazem', '5/5/2002', 'Miami'), (1465139, 'Heba', 'Ezz', '9/23/2002', 'Sedi Gaber'),  
(1465140, 'Abdelrahman', 'Ashraf', '3/6/2004', 'Victoria'), (1465141, 'Sara', 'Hazem', '12/10/2004', 'Miami'),  
(1465142, 'omar', 'Hazem', '3/15/2004', 'Gleem'), (1465144, 'Abdelrahman', 'Hazem', '1/16/2004', 'Miami'),  
(1465146, 'Nada', 'Tarek', '7/1/2004', 'Sidi Beshr'), (1465147, 'Abdullah', 'Ehab', '2/17/2004', 'Roushdy'),  
(1465148, 'Ahmed', 'Hazem', '3/19/2004', 'Gleem'), (1465149, 'Rehab', 'Khaled', '4/18/2004', 'Sapapasha'),  
(1465150, 'Mohamed', 'samir', '8/10/2004', 'Gleem'), (1465151, 'Mai', 'Ehab', '9/26/2004', 'Smouha'),  
(1465152, 'Ehab', 'Mahmoud', '9/23/2004', 'Sidi Gaber'), (1465153, 'Mahmoud', 'Mohamed', '12/5/2004', 'Victoria'),  
(1465154, 'Mohamed', 'Barakat', '6/13/2004', 'Miami')
```



## Output:

ID	First_Name	Last_Name	Date_of_Birth	Address
1465124	Nureen	Barakat	7/29/2003	Sababasha
1465125	Yassimen	Mohamed	4/5/2003	Smouha
1465126	Noha	Mohamed	6/1/2003	Smouha
1465127	Doha	Mohamed	5/23/2003	Smouha
1465128	Reem	Mohamed	2/25/2003	Sababasha
1465129	Iana	Mohamed	6/25/2003	Sababasha
1465130	Merna	Barakat	9/3/2002	Sababasha
1465131	Sherif	Mohamed	10/3/2002	Smouha
1465132	Hesham	Mohamed	10/2/2002	Sababasha
1465133	Amr	Mohamed	10/23/2002	Sababasha
1465134	Mohamed	Medhat	9/23/2002	roushdy
1465135	Ahmed	Medhat	7/25/2002	roushdy
1465136	Ahmed	Medhat	8/6/2002	Miami
1465137	Mohamed	Ezz	11/15/2002	Sedi Gaber
1465138	Fatma	Hazem	5/5/2002	Miami
1465139	Heba	Ezz	9/23/2002	Sedi Gaber
1465140	Abdelrahman	Ashraf	3/6/2004	Victoria
1465141	Sara	Hazem	12/10/2004	Miami
1465141	Sara	Hazem	12/10/2004	Miami
1465142	omar	Hazem	3/15/2004	Gleem
1465143	Nadine	Tarek	12/8/2004	Sidi Beshr
1465144	Abdelrahman	Hazem	1/16/2004	Miami
1465145	Menna	Ehab	6/6/2004	Sidi Beshr
1465146	Nada	Tarek	7/1/2004	Sidi Beshr
1465147	Abdullah	Ehab	2/17/2004	Roushdy
1465148	Ahmed	Hazem	3/19/2004	Gleem
1465149	Rehab	Khaled	4/18/2004	Sapapasha
1465150	Mohamed	samir	8/10/2004	Gleem
1465151	Mai	Ehab	9/26/2004	Smouha
1465152	Ehab	Mahmoud	9/23/2004	Sidi Gaber
1465153	Mahmoud	Mohamed	12/5/2004	Victoria
1465154	Mohamed	Barakat	6/13/2004	Miami



- ❖ Count number of students we have by count number of IDs using  
**SELECT COUNT (ID) FROM table\_name**

```
SELECT COUNT(ID) FROM student
```

**Output:**

COUNT(ID)
31

## 5) Create course table using **CREATE TABLE**

- Inside it we will put the name of each column and its data type
- We will make ID column primary key of department table using **PRIMARY KEY** as each ID is unique
- We will put department ID as foreign key using **FOREIGN KEY**(any name of column we want) **REFERENCES** table we take its primary key as foreign key of course table (name of primary key column)  
instructor ID is the foreign key of course table as relation between these 2 tables is one to many so we take primary key of **one** relation side (instructor) to be foreign key in **many** relation side (course)

```
CREATE Table course(ID int PRIMARY KEY, Name varchar(30), Credits int , Prerequisites int , Instructor_ID int ,  
FOREIGN KEY(Instructor_ID) REFERENCES instructor(ID))
```

- Insert values in course table using **INSERT INTO table\_Name**, and it takes name of each column, and then put values inside each column using **VALUES**:

```
INSERT INTO course(ID , Name , Credits , Prerequisites , Instructor_ID) VALUES(00101, 'Linear Algebra',3,null,2),  
(00102, 'Calculus',3,null,9),(00103, 'Introduction to Computer Systems',3,null,10),  
(00104, 'Introduction to Data Sciences',3,null,7),(00105, 'ProgrammingI',3,null,3),  
(00106, 'Probability and StatisticsI',3,null,8),(00107, 'Discrete Structures',3,null,4),  
(00108, 'Data Structures and Algorithms',3,00105 ,11),  
(00109, 'Introduction to Artificial Intelligence',3,00103 ,15),  
(00110, 'ProgrammingII',3,00105,16),(00201,'Probability and StatisticsII',3,00106,17),  
(00202, 'Introduction to Databases',3,00105,null),(00203, 'Numerical Computations',3,00101,12),  
(00204, 'Cloud Computing',3,00101,null), (00205, 'Machine Learning',3,00109,null),  
(00206, 'Data Mining and Analytics',3,00201,null),(00303, 'Algorithm Design',3,00108,13),  
(03202, 'Operations Research',3,00106,6),(03203, 'Pattern Recognition',3,00106,5),  
(03204, 'Neural Networks',3,00109,1),(0001, 'Critical Thinking',2,null,19),  
(0002, 'Innovation and Leadership',2,null,20),  
(100, 'Principal of Management',2,null,18),(0005, 'Economy Science',2,null,21)
```

## Output:

ID	Name	Credits	Prerequisites	Instructor_ID
1	Critical Thinking	2	NULL	19
2	Innovation and Leadership	2	NULL	20
5	Economy Science	2	NULL	21
100	Principal of Management	2	NULL	18
101	Linear Algebra	3	NULL	2
102	Calculus	3	NULL	9
103	Introduction to Computer Syste	3	NULL	10
104	Introduction to Data Sciences	3	NULL	7
105	ProgrammingI	3	NULL	3
106	Probability and StatisticsI	3	NULL	8
107	Discrete Structures	3	NULL	4
108	Data Structures and Algorithms	3	105	11
109	Introduction to Artificial Int	3	103	15
110	ProgrammingII	3	105	16
201	Probability and StatisticsII	3	106	17
202	Introduction to Databases	3	105	NULL
203	Numerical Computations	3	101	12
204	Cloud Computing	3	101	NULL
205	Machine Learning	3	109	NULL
206	Data Mining and Analytics	3	201	NULL
303	Algorithm Design	3	108	13
3202	Operations Research	3	106	6
3203	Pattern Recognition	3	106	5
3204	Neural Networks	3	109	1

- ❖ Count number of courses we have by count number of Names using **SELECT COUNT (Name) FROM table\_name**

```
Select COUNT(Name) FROM course
```

**Output:**

COUNT(Name)
24

- ❖ Count number of courses who has instructor to teach it to the students by count number of Instructor\_ID using **SELECT COUNT (Instructor\_ID) FROM table\_name**

```
Select COUNT(Instructor_ID) FROM course
```

**Output:**

COUNT(Instructor_ID)
20

## 6) Create enroll table using **CREATE TABLE**

- Inside it we will put the name of each column and its data type  
We make enroll as entity (table) as the relation between student and course is **many to many** so the relation between them(enroll) must be different entity
- We will put student ID and course ID as foreign key using **FOREIGN KEY**(any name of column we want) **REFERENCES** table we take its primary key as foreign key of enroll table (name of primary key column)  
student ID and course ID are the foreign key of course table as relation between student and course tables is **many to many** so we take primary key of each of them as foreign key in enroll table

```
CREATE Table enroll(Student_ID int , Course_ID int, Year int, Semester int, Grade float , Total_Credits int ,  
FOREIGN KEY(Student_ID) REFERENCES student(ID) , FOREIGN KEY(Course_ID) REFERENCES course(ID))
```

- Insert values in enroll table using **INSERT INTO table\_Name**, and it takes name of each column, and then put values inside each column using **VALUES**:

```
INSERT INTO enroll(Student_ID, Course_ID , Year , Semester , Grade , Total_Credits) VALUES  
(1465124, 00101, 2023, 4, 3.51, 56), (1465125, 00102, 2023, 2, 2.5, 17), (1465126, 00103, 2023, 2, 3, 17),  
(1465127, 00102, 2023, 4, 3.2, 53), (1465128, 00105, 2023, 6, 2.3, 102), (1465129, 03202, 2023, 2, 3.3, 17),  
(1465130, 00109, 2023, 4, 1.95, 48), (1465131, 00203, 2023, 6, 3.14, 100), (1465132, 00110, 2023, 2, 3.66, 19),  
(1465133, 00105, 2023, 2, 3.98, 19), (1465134, 03202, 2023, 4, 2.72, 56), (1465135, 00102, 2023, 2, 3.64, 17),  
(1465136, 00103, 2023, 2, 2.53, 17), (1465137, 00101, 2023, 2, 3.92, 17), (1465138, 00109, 2023, 2, 3.39, 17),  
(1465139, 00110, 2023, 4, 3.72, 53), (1465140, 00101, 2023, 2, 3.22, 17), (1465141, 00109, 2023, 4, 1.86, 48),  
(1465142, 00102, 2023, 2, 3.48, 17), (1465143, 03202, 2023, 6, 3.9, 108), (1465144, 00105, 2023, 2, 4, 19),  
(1465145, 00203, 2023, 4, 3.68, 53), (1465146, 00105, 2023, 2, 2.73, 17), (1465147, 00103, 2023, 2, 3.58, 17),  
(1465148, 00110, 2023, 2, 3.67, 17), (1465149, 03202, 2023, 6, 3.32, 102), (1465150, 00102, 2023, 2, 2.69, 17),  
(1465151, 00110, 2023, 2, 3.44, 17), (1465152, 00105, 2023, 4, 3.73, 53), (1465153, 00102, 2023, 2, 2, 17),  
(1465154, 03203, 2023, 6, 3.8, 106)
```

## Output:

Student_ID	Course_ID	Year	Semester	Grade	Total_Credits
1465124	101	2023	4	3.51	56
1465125	102	2023	2	2.5	17
1465126	103	2023	2	3	17
1465127	102	2023	4	3.2	53
1465128	105	2023	6	2.3	102
1465129	3202	2023	2	3.3	17
1465130	109	2023	4	1.95	48
1465131	203	2023	6	3.14	100
1465132	110	2023	2	3.66	19
1465133	105	2023	2	3.98	19
1465134	3202	2023	4	2.72	56
1465135	102	2023	2	3.64	17
1465136	103	2023	2	2.53	17
1465137	101	2023	2	3.92	17
1465138	109	2023	2	3.39	17
1465139	110	2023	4	3.72	53
1465140	101	2023	2	3.22	17
1465141	109	2023	4	1.86	48
1465142	102	2023	2	3.48	17
1465143	3202	2023	6	3.9	108
1465144	105	2023	2	4	19
1465145	203	2023	4	3.68	53
1465146	105	2023	2	2.73	17
1465147	103	2023	2	3.58	17
1465148	110	2023	2	3.67	17
1465149	3202	2023	6	3.32	102
1465150	102	2023	2	2.69	17
1465151	110	2023	2	3.44	17
1465152	105	2023	4	3.73	53
1465153	102	2023	2	2	17
1465154	3203	2023	6	3.8	106

- ❖ select students who has grade>=3 using **SELECT \* FROM table\_name WHERE Grade>=3** and rank them by grade in descending order using **GROUP by Grade ORDER by Grade DESC**

**Note:** use \* as we want to show all attributes (columns) of student who has enroll a course and his grade>=3

```
SELECT * FROM enroll WHERE Grade>=3 GROUP by Grade ORDER by Grade DESC
```

## Output:

Student_ID	Course_ID	Year	Semester	Grade ▾ 1	Total_Credits
1465144	105	2023	2	4	19
1465133	105	2023	2	3.98	19
1465137	101	2023	2	3.92	17
1465143	3202	2023	6	3.9	108
1465154	3203	2023	6	3.8	106
1465152	105	2023	4	3.73	53
1465139	110	2023	4	3.72	53
1465145	203	2023	4	3.68	53
1465148	110	2023	2	3.67	17
1465132	110	2023	2	3.66	19
1465135	102	2023	2	3.64	17
1465147	103	2023	2	3.58	17
1465124	101	2023	4	3.51	56
1465142	102	2023	2	3.48	17
1465151	110	2023	2	3.44	17
1465138	109	2023	2	3.39	17
1465149	3202	2023	6	3.32	102
1465129	3202	2023	2	3.3	17
1465140	101	2023	2	3.22	17
1465127	102	2023	4	3.2	53
1465131	203	2023	6	3.14	100
1465126	103	2023	2	3	17



- ❖ Get the inner intersection between student and enroll tables using  
**SELECT column\_names FROM student INNER JOIN enroll on  
bond\_between\_the\_2tables**

```
SELECT First_Name, Last_Name, Student_ID, Course_ID, Semester, Grade, Total_Credits FROM
student INNER JOIN enroll on enroll.Student_ID=student.ID
```

## Output:

First_Name	Last_Name	Student_ID	Course_ID	Semester	Grade	Total_Credits
Nureen	Barakat	1465124	101	4	3.51	56
Yassimen	Mohamed	1465125	102	2	2.5	17
Noha	Mohamed	1465126	103	2	3	17
Doha	Mohamed	1465127	102	4	3.2	53
Reem	Mohamed	1465128	105	6	2.3	102
Iana	Mohamed	1465129	3202	2	3.3	17
Merna	Barakat	1465130	109	4	1.95	48
Sherif	Mohamed	1465131	203	6	3.14	100
Hesham	Mohamed	1465132	110	2	3.66	19
Amr	Mohamed	1465133	105	2	3.98	19
Mohamed	Medhat	1465134	3202	4	2.72	56
Ahmed	Medhat	1465135	102	2	3.64	17
Ahmed	Medhat	1465136	103	2	2.53	17
Mohamed	Ezz	1465137	101	2	3.92	17
Fatma	Hazem	1465138	109	2	3.39	17
Heba	Ezz	1465139	110	4	3.72	53
Abdelrahman	Ashraf	1465140	101	2	3.22	17
Sara	Hazem	1465141	109	4	1.86	48
omar	Hazem	1465142	102	2	3.48	17
Nadine	Tarek	1465143	3202	6	3.9	108
Abdelrahman	Hazem	1465144	105	2	4	19
Menna	Ehab	1465145	203	4	3.68	53
Nada	Tarek	1465146	105	2	2.73	17
Abdullah	Ehab	1465147	103	2	3.58	17
Ahmed	Hazem	1465148	110	2	3.67	17
Rehab	Khaled	1465149	3202	6	3.32	102
Mohamed	samir	1465150	102	2	2.69	17
Mai	Ehab	1465151	110	2	3.44	17
Ehab	Mahmoud	1465152	105	4	3.73	53
Mahmoud	Mohamed	1465153	102	2	2	17
Mohamed	Barakat	1465154	3203	6	3.8	106

## 7) Create course table using **CREATE TABLE**

- Inside it we will put the name of each column and its data type
- We will make Number column primary key of department table using **PRIMARY KEY** as each ID is unique
- We will put course ID as foreign key using **FOREIGN KEY**(any name of column we want) **REFERENCES** table we take its primary key as foreign key of classroom table (name of primary key column)  
course ID is the foreign key of classroom table as relation between these 2 tables is one to many so we take primary key of **one** relation side (course) to be foreign key in **many** relation side (classroom)

```
CREATE Table classroom(Number int PRIMARY KEY, Capacity int , Course_ID int ,  
                        FOREIGN KEY(Course_ID) REFERENCES course(ID))
```

- Insert values in classroom table using **INSERT INTO table\_Name** , and it takes name of each column, and then put values inside each column using **VALUES**:

```
INSERT INTO classroom(Number, Capacity, Course_ID) VALUES  
(420,60,101),(425,60,102),(406,40,103),(404,60,104),(402,60,105),  
(415,30,106),(417,30,107),(304,60,108),(305,60,109),(306,30,110),  
(307,30,201),(308,30,202),(101,20,203),(102,20,303),(201,50,100),  
(202,50,1),(207,50,2),(301,70,5),(302,70,3203),(303,70,3202)
```

## Output:

Number	Capacity	Course_ID
101	20	203
102	20	303
201	50	100
202	50	1
207	50	2
301	70	5
302	70	3203
303	70	3202
304	60	108
305	60	109
306	30	110
307	30	201
308	30	202
402	60	105
404	60	104
406	40	103
415	30	106
417	30	107
420	60	101
425	60	102

- ❖ Count number of classrooms available for the courses using **SELECT COUNT (Number) FROM table\_name**

```
Select COUNT(Number) FROM classroom;
```

### Output:

COUNT(Number)
20

- ❖ Select Number and Capacity of classrooms whose capacity>30 using **SELECT Number,Capacity FROM table\_name WHERE Capacity>30**

```
SELECT Number,Capacity FROM classroom WHERE Capacity>30
```

### Output:

Number	Capacity
201	50
202	50
207	50
301	70
302	70
303	70
304	60
305	60
402	60
404	60
406	40
420	60
425	60