1) Brief discussion: -

To design ABCE (ABC Education) database, firstly I read the case study and understand the requirements those mention in given case study. After that I make ERD in 3NF on paper and fulfill all the requirements. Then ERD contains all entities, attributes and relationship between entities those mention in the case study. When I satisfy with ERD then I create ERD in Draw.io in Crown foot model.

- **Problem: -**

I face problem when I create ERD because this ERD contains lots of entities and attributes. I face difficult when I joins the entities means to create the relationship between them. Another issue when I draw ERD in Draw.io. Due to many entities it was difficult to show all entities in small space and manage relationship between then this tasks consume lots of time. But at last I manage all these things and design ERD in proper manner.

2) Business rules: -

<u>Entity: Department</u>
Each Department has one or many College
    Each Department has one to many College_Department.
    Each College_Department recode is related to one Department.
Each Department has one or more faculty member.
Each Department has one or more Course.

<u>Entity: College</u>
Each college has one or more Department
    Each college has one to more College_Department.
    Each College_Department recode is related to one College.
Each College has one and only one Administrative_Staff
Each Collage has one or many Machine.

<u>Entity: Staff_Info</u>
Each Staff_Info has one and only one Administrative_Staff
Each Staff_Info has one or one and only one Faculty.

Entity: Administrative_Staff
Each Administrative_Staff has one and only one College.
Each Administrative_Staff has one and only one Staff_Info
Each Administartive_Staff may or may not have a Machine

Entity: Faculty
Each Faculty has one and only one Staff_Info
Each Faculty has one or more Department
Each faculty has one and only one Collage_Department.
Each Department has one or many College_Department.
Each Faculty has one and only one Collage
Each Faculty has one and only one Machine.
Each Faculty has one or more Class_Detail.

Entity: Machine
Each Machine has zero or many Administrative_Staff
Each Machine has one and only one College.
Each Machine has zero or more Student_Info
Each Machine has zero or more Enrollment
Each Enrollment has recode is related to a Student_Info
Each Machine has zero or more Faculty

Entity: Course
Each Course has one or many Subject
Each Course has one and only one Department.

Entity: Subject
Each Subject has one or more Text_Book
Each Subject has one or more Class_Detail
Each Subject has one or many Summary

Entity: Text_Book
Each Text_Book has one and only on Subject.

Entity: Student_Info
Each Student_Info has one or more Enrollment.
Each Student_Info has one and many Request_Form
Each student_Info has one and only one Report_Class

Entity: Report_Class
Each Report_Class has one and only one Summary.
Each Report_Class has one and only one Student_Info

Entity: Summary

Each Summary has one and only one Report_Class
Each Summary has one and only one Subject

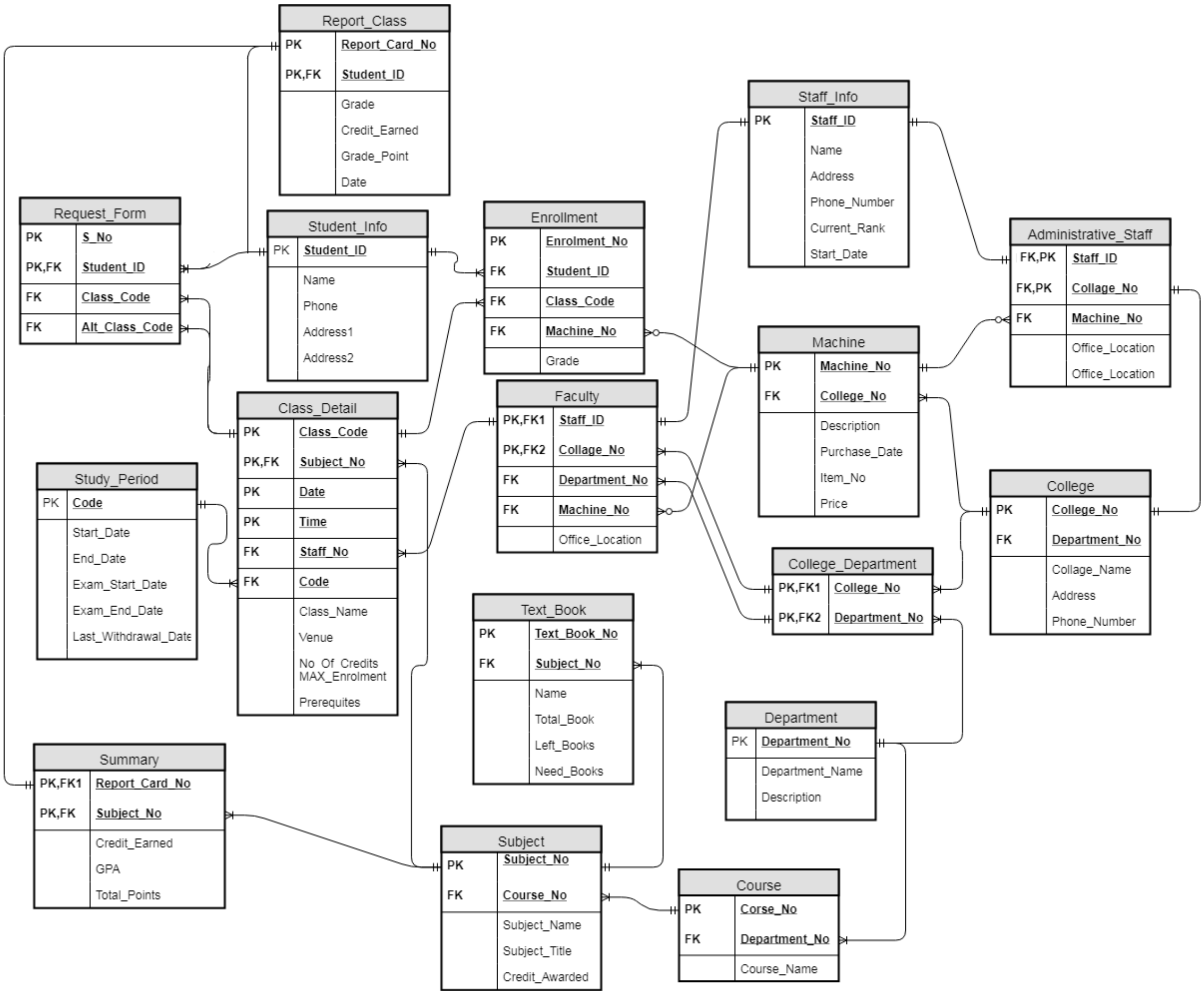
Entity: Study_Period
Each Study_Period has one and only one Class_Detial

Entity: Class_Detail
Each Class_Detail has one and only many Enrollment
Each Class_Detail has one and only many Request_Form
Each Class_Detail has one and only many Alternative_Reguest_Form
Each Class_Detail has one and only one Subject
Ecah Class_Detail has one and only one Faculty
Each Class_Detail has one and only one Study_Period

Entity: Enrollment
Each Enrollment has one and only one Student_Info
Each Enrollment has one and only one Class_Detail
Each enrollment has zero or one Machine

Entity: Request_Form
Each Request_Form has one and only one student_Info
Each Request_Form has one and only one Class_Detail
Each Request_Form has one and only one Altrative_Class_Detail

3) Entity relationship diagram (ERD) :-



4) Summary: -

- **Assumption: -**

  I design ERD according to case study. I just create three extra tables to maintain the relationship between the entity and these are following: -

  1) College_Department table: - there are many to many relationships between the Department and College. To maintain the relationship I create CollageDepartment table between them.
  2) Staff_Info table: -there are two types of staff one is Administrative_Staff and other one is Faculty staff. To maintain the information of those staff I create comman table Staff_Info.

3) Enrollment table: - to maintain the student grade, machine related to student and class_Detail of a student I create Enrollment table.

- ## Justifications: -

1) **Department: -**
    - Primary key: - Department_No
    - Foreign key: - there is no foreign key
    - Relationship: - 1:M between Department and College_Department.
    - Cardinality: -one Department many College_Department and each College_Department as one Department.
    - Optionality: - each Department has one College_Department and each College_Department has one Department.

s. no	Attribute name	Data type	Data domain
1	Department_ID	Varchar	3
2	Department_Name	varchar	50
3	Description	varchar	255

2) **College: -**
    - Primary key: -College_No
    - Foreign key: -Department_No
    - Relationship: - 1:m, each College has one or more Collge_Department and each College_Department has one and only one College.
    - Cardinality: - one College has many College_Department and one College_Department has one College.
    - Optionality: - one College has one College_Department and one College_Department has one College.

s. no	Attribute name	Data type	Data domain
1	College_No	varchar	20
2	Department_No	varchar	3
3	College_name	Varchar	100
4	Address	varchar	200
5	Phone_Number	varchar	20

3) **College_Department: -**
    - Primary key: - College_No, Department_No
    - Foreign key: -College_No, Department_No
    - Relationship: -
        - 1:M between Department and College_Department.

- 1:m, each College has one or more Collge_Department and each College_Department has one and only one College.

- Cardinality: -
  - one Department many College_Department and each College_Department as one Department.
  - one College has many College_Department and one College_Department has one College.
- Optionality: -
  - one College has one College_Department and one College_Department has one College.
  - each Department has one College_Department and each College_Department has one Department.

s. no	Attribute name	Data type	Data domain
1	College_No	varchar	10
2	Department_No	varchar	3

4) Staff_Info: -

- Primary key: - Staff_ID
- Foreign key: - there is no foreign key.
- Relationship: -
  - each Staff_Info has one and only one Administrative_Staff and each Administrative_Staff has one and only one Staff_Infp.
  - Each Staff_Info has one and only one Faculty and each Faculty has one and only one Staff_Info.
- Cardinality: -
  - one Staff_Info has one Administrative_Staff and one Administrative Staff has one Staff_Info.
  - One Staff_Info has one Faulty and one Faulty has one Staff_Info.
- Optionality: -
  - one Staff_Info has one Administrative_Staff and one Administrative Staff has one Staff_Info.
  - One Staff_Info has one Faulty and one Faulty has one Staff_Info.

s. no	Attribute name	Data type	Data domain
1	Staff_ID	varchar	10
2	Name	varchar	50
3	Address	varchar	200
3	Phone_Number	varchar	20
4	Current_Rank	varchar	20
5	Start_Date	date	-

5) **Administrative_Staff: -**

- Primary key: -Staff_ID, College_No
- Foreign key: -College_No, Machine_No
- Relationship: -
  - Each Administrative_Staff has one and only one College and each College has one and only one Admimistrative_Staff.
  - Each Administrative_Staff has one and only one Staff_Info and each Staff_Info has one and only one Administrative_Staff.
  - Each Administrative_Staff has one and only one Machine and each Machine has Zero or many Administrative_Staff.
- Cardinality: -
  - Each Administrative_Staff has one College and each College has one Admimistrative_Staff.
  - Each Administrative_Staff has one Staff_Info and each Staff_Info has one Administrative_Staff.
  - Each Administrative_Staff has one Machine and each Machin has many Administrative_Staff.
  -
- Optionality: -
  - Each Administrative_Staff has one College and each College has one Administrative_Staff.
  - Each Administrative_Staff has one Staff_Info and each Staff_Info has one Administrative_Staff.
  - Each Administrative_Staff has one Machine and each Machine has Zero Administrative_Staff.

s. no	**Attribute name**	**Data type**	**Data domain**
1	Staff_ID	varchar	10
2	College_No	varchar	10
3	Machine_No	varchar	10
4	Office_Location	varchar	100

6) **Faculty: -**

- Primary key: - Staff_ID, College_No
- Foreign key: - Staff_ID, College_No, Department_No, Machine_No
- Relationship: -
  - Each Faculty has one and only one Staff_Info and each Staff_Info has one and only one Faculty.
  - Each Faculty has one and only one College_Department and each College_Department has one many many Faculty.

  - Each Faculty has one and only one Machine and each Machine has Zero or many Faculty.

- Cardinality: -
  - Each Faculty has one Staff_Info and each Staff_Info has one Faculty.
  - Each Faculty has one College_Department and each College_Department has many Faculty.
  - Each Faculty has one Machine and each Machine has many Faculty.

- Optionality: -
  - Each Faculty has one Staff_Info and each Staff_Info has one Faculty.
  - Each Faculty has one College_Department and each College_Department has one Faculty.
  - Each Faculty has one Machine and each Machine has Zero Faculty.

s. no	Attribute name	Data type	Data domain
1	Staff_ID	varchar	10
2	College_No	varchar	10
3	Department_No	varchar	3
4	Machine_No	varchar	10
5	Office_Location	varchar	100

7) Machine: -

- Primary key: - Machine_No
- Foreign key: - College_No
- Relationship: -
  - Each Machine has zero or many Administartive_Satff and each Administrative_Staff has one and only one Machine.
  - Each Machine has one and only one College and each College has one and many Machine.
  - Each Machine has zero or more Enrollment and each Enrollment has one and only one Machine.
  - Each Machine has Zero or many Faculty and Each Faculty has one and only one Machine.
- Cardinality: -
  - Each Machine has many Administartive_Satff and each Administrative_Staff has one Machine.
  - Each Machine has one College and each College has many Machine.
  - Each Machine more Enrollment and each Enrollment has one Machine.

- Each Machine has many Faculty and Each Faculty has one Machine.

- Optionality: -
    - Each Machine has zero Administartive_Satff and each Administrative_Staff has one Machine.
    - Each Machine has one College and each College has one Machine.
    - Each Machine has zero Enrollments and each Enrollment has one Machine.
    - Each Machine has Zero Faculty and Each Faculty has one Machine.

s. no	Attribute name	Data type	Data domain
1	Machine_No	varchar	10
2	College_No	varchar	10
3	Description	varchar	200
4	Purchase_Date	date	-
5	Item_No	varchar	10
6	Price	double	-

4) Special Cases: -

- There are many to many relationships between the Department and College. To maintain the relationship I create Collage_Department table between them.
- To maintain the student grade, machine related to student and class_Detail of a student I create Enrollment table.

Data integrity issues:-

- **Physical data integrity: -** when user try to fetch the student report card summary then it may create issue to get correct result. We need also create venue table to specify each class venue.

- **Logical data integrity: -** it creates issue if user input null foreign key value in database. Admin set some rule mange this database if users don’t follow those rules in it also create issue.
- **Domain constrains: -** when table is create in the database in programmer set the domain constrains for each attribute of table. if programmer set integer for primary key and user try to enter character then it create issue.