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.

1. **Business rules: -**

Entity: Department

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.

Entity: College

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.

 Entity: Staff\_Info

 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

1. **Entity relationship diagram (ERD) :-**



1. **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 |

1. **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 |

1. **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 |

1. **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 | - |

1. **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 |

1. **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 |

1. **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 | - |

1. **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.