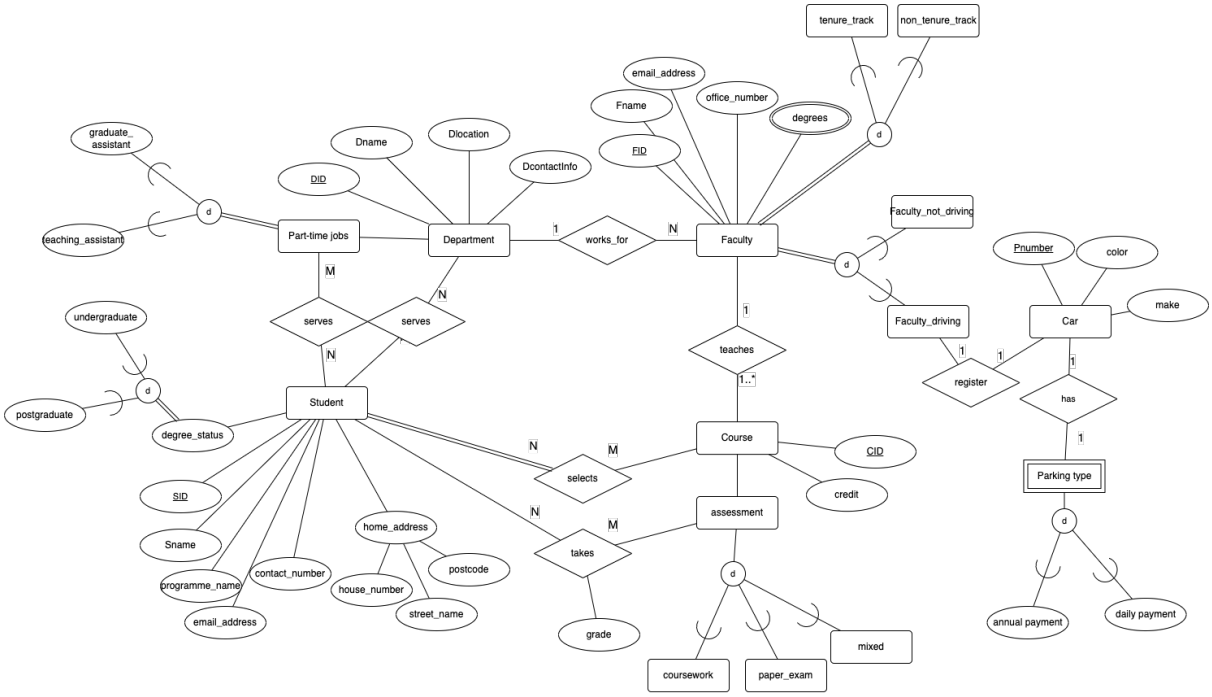
Problem 1:



Problem 2:

1. the strong entities:

Publisher (URL) Publisher (URL, Pname, Paddress)
Library_Item (IID) Library_Item (IID, Title)
Borrower (BID) Borrower (BID, Bname)

2. the weak entity:

without any weak entity in this case.

3. one-to-one (1:1) relationship:

without any one-to-one relationship types in this case.

4. one-to-many (1:N) relationship:

Since Publishes and Borrow are both one-many relationships, it does not require a table. We just need to put the primary key from many-side into the 1-side, as foreign keys.

Publishes Publisher (URL, Pname, Paddress, IID*)
Borrow Borrower (BID, Bname, IID*, dates)

5. many-to-many (N:M) relationship:

Since Reserved and Write are both many-to-many relationship, creating a new table and pulling primary keys from others as composite primary key.

Write Write (Bnumber*, AID*, URL*)
Reserved Reserved (BID*, IID*, Order_ID*)

6. Mapping of multivalued attributes

Phone_numbers Publisher_phone_numbers (PNID*, Phone_number)

7. Mapping of N-ary relationship types

without N-ary relationship types

8. Options for mapping Specialization or Generalization:
Borrower (BID, Bname, Staff, Student)
Staff (BID*)
Student (BID*)

allows to store more information in the future about public sector
Publisher (URL, Pname, Paddress, Publisher_type)
Public_sector (Publisher_type*)
Private_sector (Publisher_type*)

Library_Item (IID, Title)
Book (IID*)
Journal (IID*, date, Issue_no, Volume)
Magazine (IID*, date, Issue_no)
Conference_proceedings (IID*, date, Location)

Collecting all the tables together:

Borrower (BID, Bname, IID*, dates)
Write (Bnumber*, AID*, URL*)
Reserved (BID*, IID*, Order_ID*)
Publisher_phone_numbers (PNID*, Phone_number)
Authors (AID)
Borrower (BID, Bname, Staff, Student)
Staff (BID*)
Student (BID*)
Publisher (URL, Pname, Paddress, Publisher_type, IID*)
Public_sector (Publisher_type*)
Private_sector (Publisher_type*)
Library_Item (IID, Title)
Book (IID*)
Journal (IID*, date, Issue_no, Volume)
Magazine (IID*, date, Issue_no)
Conference_proceedings (IID*, date, Location)

Problem 3:

3.1
table Library_Item, it allows the library to store more type of items such as CDs, Newspaper and so on. However, it causes too many tables.
table Borrower, it allows the library to save more borrower, even though there are normally 2 type of people, neither staff or student. it might cause too many tables as well.
table Publisher, we can see that we have 2 higher level entities which are public sector and private sector, each one may have more data itself.

3.2

The table Book is a good candidate for horizontal partitioning.

IID	Bname	Desc
1	a	
2	b	
3	c	
4	d	

Horizontal Partition(Hash based)

IID	Bname	Desc
1	a	
3	c	

User table-Shard1

IID	Bname	Desc
2	b	
4	d	

User table-Shard2

There are normally many books. This partition reduces the speed of reading operations.
select * from Book where IID = 1 and IID = 3;
select * from Book where IID = 2 and IID = 4;

3.3

Table Publisher is a good candidate for vertical partitioning

URL	Pname	Paddress	Publisher_type
url1	PN1	addr1	public
url2	PN2	addr2	private
url3	PN3	addr3	public
url4	PN4	addr4	private

Vertical Partition(Range based)

URL	Pname	Paddress
url1	PN1	addr1
url2	PN2	addr2
url3	PN3	addr3
url4	PN4	addr4

Uer table

URL	Publisher_type
url1	public
url2	private
url3	public
url4	private

Extension table

Sometimes, a few basic information of publisher might be checked frequently, or we don't have to know the certain part out of user information.
select URL, Pname, Paddress from Publisher;

Problem 4:

4.1

$U_R = \{ F \}$
$N = U - U_R = \{ A, B \}$
$AB^+ = \{ A, B, F \} = R$
Thus, N is a minimal key. The candidate key is { A, B }.

4.2

relation R is not in BCNF. R is atomic so it is in the 1NF. B is partial of AB so there is a partial dependency cause that it is not in the 2NF.
To be BCNF, it must be a 3NF and for any dependency A--->B, A should be a super key.

let D be the set of all attributes of R
D = { A, B, F }
B --> F
D = { A, B} => Rnew = { F, B }
It is now in a collection of BCNF relations.