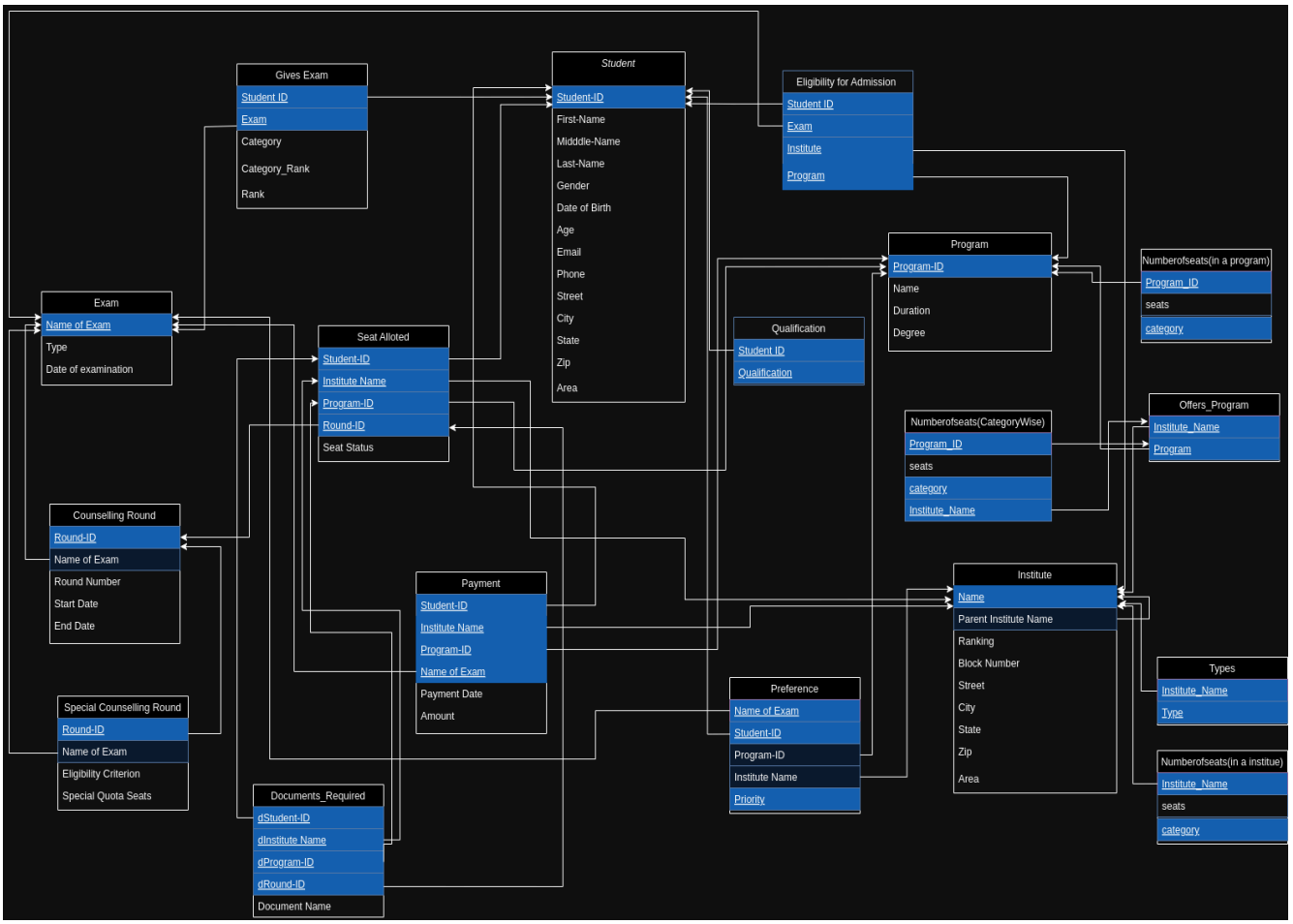
Project Phase 3-Team 20

Converting ERDiagram into Relational Model

- **Mapping of Regular Entity Types**. For each strong entity type E in the ER schema, we created a relation R that includes all the simple attributes of E and the composite attributes were converted into simple component attributes for example in student entity ,the composite attribute address was broken down into the simple component attributes like zipcode, state, country , street etc.
- **Mapping of Weak Entity Types**. For each weak entity type W in the ER schema with owner entity type E,we created a relation R and included all simple attributes and the composite attribute was broken down into simple component attributes of W as attributes of R. Also we made sure to include the foreign key attributes of R, that is the primary key attributes of the relations that correspond to the owner entity types. This took care of mapping the identifying relationship type of W. The primary key of R is the combination of the primary keys of the owners and the partial key of the weak entity type W.
- **Mapping of Binary 1:1 Relationship Types**. We didn't have any Binary 1:1 relationships so we didn't have to take care of this case.
- **Mapping of Binary 1:N Relationship Types.** For this we followed the foreign key approach. For each regular binary 1:N relationship type R,identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R. This is what we followed for the relationship “Is for”
- **Mapping of Binary M:N Relationship Types.** For each binary M:N relationship type R,we made a new relation S to represent R and included as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. We also made sure to include all the relationship attributes in this table . We did the above procedure for relationships “Offers Program” and “Gives Exam”.
- **Mapping of Multivalued Attributes**. For each multivalued attribute A, we made a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as a multivalued attribute. The primary key of R is the combination of A and K. We did this for MVA’s like “Qualification” in Student relation , “Type” in Institute relation etc.
- **Mapping of N>2 Relationship Types.** For each n>2 relationship type R, we made a new relationship relation S to represent R. We made sure to include as foreign key attributes in S the primary keys of the relations that represent the participating entity types and we also included simple attributes of the n>2 relationship type as attributes of S. The primary key of S was chosen to be a combination of all the foreign keys that reference the relations representing the participating entity types whose cardinality constraints on any of the entity types E participating in R is not 1. We applied this for n>2 relationships like “Payment” , “Eligibility” etc.

- **Mapping of Subclass.** For this we made a new relation R and made sure it includes the primary key of the main type . In our case the primary key of the subclass turned out to be the same as that of the main relation.We applied this approach while making a new relation of Subclass “Special Counselling Round”.

Using this steps the Relationship model turned out to be



- We have also made some minor changes in the cardinality of some the relationships like in “Payment” (Student:Institute:Program:Exam = N:L:M:K) and “Seat Alloted” (Student:Institute:Program:Counselling Round = N:L:M:K). These were the only changes that we made
- We also removed “total number of seats” derived attribute from the relations Institute,Program and Offers Function , as it can be calculated using SUM function , also removing this attribute reduced redundancy which is also beneficial for us and also the use of the data in total number of seats was not frequent so removing it seemed to be a better option for us.
- The attribute “Exams given by student” was removed from the “Student” relation as it can be obtained from the “Gives Exam” relation.

- Here in Students_Address and Institute_Address relation we added a attribute “Area”.

Converting Relational Model to 1NF

- This form does not allow Multivalued attributes , Composite attributes and nested relations.
- We already took care of Multivalued attributes by making a new table for them in order to reduce redundancy and we also converted composite attributes to simple component attributes. So as a result both of these were taken care of in the relational model itself.
- Any of the relations in the relational model generated from the ER Diagram did not have nested relations.
- As a result the 1NF form turned out to be the same as the Relational Model.

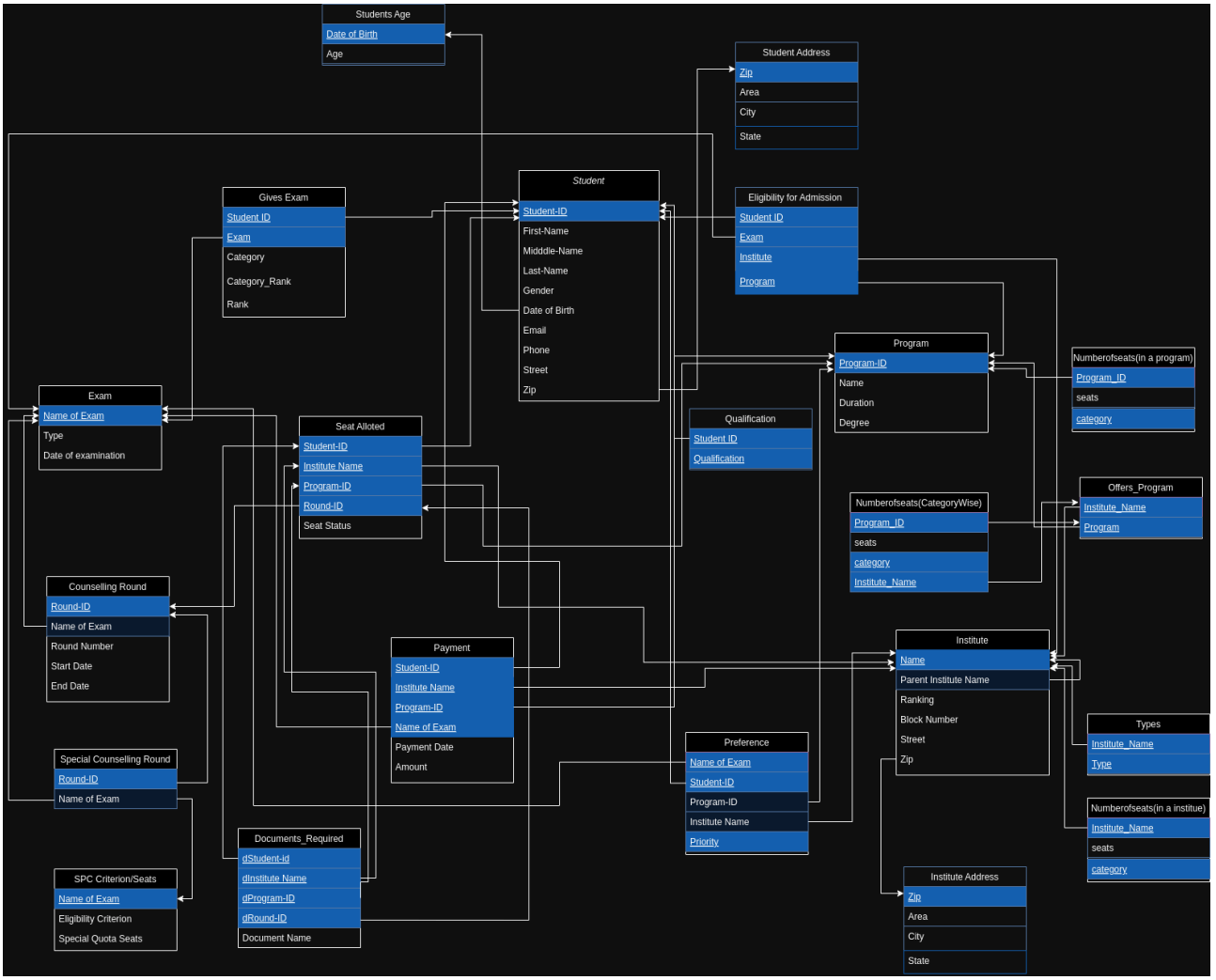
Converting 1NF to 2NF

- In this form we needed to ensure that in every relation R , every non prime attribute A in R was fully functionally dependent on the primary key K.
- We checked each and every relation in the 1NF form , and checked if there was any partial dependency.
- We found out that every non prime attribute A in R was already fully functionally dependent on the primary key K and there was no partial dependency.
- As a result the 2NF form turned out to be the same as the 1NF form.

Converting 2NF to 3NF

- This form does not allow transitive dependency(X->Y in R is transitive dependency, if there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key in R and both X->Z and Z->Y holds)
- We checked each and every relation in 2NF from and found out some transitive dependency in Relations Student , Institute and subclass Special Counselling Round.
- In relations Student and Institute we found that Student_id -> Zip Code and Zip code -> City,State,Area and so we made 2 new relations for this also in relation Student we another transitive dependency Student_id -> Date of Birth and Date_of_Birth -> Age and for this we made a new relation Students_Age and lastly for subclass Special Counselling Round we found one transitive dependency which was Round_id->Name of Exam and Name of Exam->Eligibility Criterion, Special Quota Seats so for this to we made a new relation named SPC Criterion/Seats.
- As a result the 3NF form turned out to be the one shown above

3NF From


Students Age
Date of Birth
Age
Student Address
Zip
Area
City
State
Gives Exam
Student ID
Exam
Category
Category_Rank
Rank
Student
Student-ID
First-Name
Middle-Name
Last-Name
Gender
Date of Birth
Email
Phone
Street
Zip
Eligibility for Admission
Student ID
Exam
Institute
Program
Program
Program-ID
Name
Duration
Degree
Numberofseats(in a program)
Program_ID
seats
category
Exam
Name of Exam
Type
Date of examination
Seat Alloted
Student-ID
Institute Name
Program-ID
Round-ID
Seat Status
Qualification
Student ID
Qualification
Numberofseats(CategoryWise)
Program_ID
seats
category
Institute_Name
Offers_Program
Institute_Name
Program
Counselling Round
Round-ID
Name of Exam
Round Number
Start Date
End Date
Payment
Student-ID
Institute Name
Program-ID
Name of Exam
Payment Date
Amount
Institute
Name
Parent Institute Name
Ranking
Block Number
Street
Zip
Types
Institute_Name
Type
Special Counselling Round
Round-ID
Name of Exam
Preference
Name of Exam
Student-ID
Program-ID
Institute Name
Priority
Numberofseats(in a institue)
Institute_Name
seats
category
Documents_Required
dStudent-id
dInstitute Name
dProgram-ID
dRound-ID
Document Name
SPC Criterion/Seats
Name of Exam
Eligibility Criterion
Special Quota Seats
Institute Address
Zip
Area
City
State