

1. Functional dependency

A functional dependency (FD) is a relationship between two attributes, typically between the PK and other non-key attributes within a table. For any relation R, attribute Y is functionally dependent on attribute X (usually the PK) this relationship is indicated by the representation below:

$$X \rightarrow Y$$

Determinant

The left side of the above FD diagram is called the determinant (X)

Dependent

The right side is the dependent(Y)

Example

The example below can be used to illustrate functional dependencies.

Student ID	Semester	Course	TA
1701	2	Physics II	Ben
1702	2	Maths II	Alice
1743	6	Visual Computer	Ted
1734	4	Numerical Methods	Katy

The functional dependencies in the table above are;

- StudentID \rightarrow (Semester, Course, TA)
Semester, Course and TA are functionally dependent on StudentID while the StudentID is the determinant of Semester, Course and TA.
- (StudentID, Course) \rightarrow TA
TA is functionally dependent on StudentID and Course while the composite (StudentID, Course) is the determinant of TA.
- (StudentID, Course) \rightarrow (Semester, TA)
Semester and TA are functionally dependent on StudentID and Course while the composite (StudentID, Course) is the determinant of Semester and TA.

2. Normalization Process

Step 1: Break Relation into Thematic Relations.

The Loan Shark List has two themes: the child and the customer information. Modification problems like inserting, updating and deleting cells can arise while editing the list. Therefore, breaking the list into two tables; Child table which include the data about the child which are the childID and ChildName, and Customer table which contains information about the customer like CustomerID, CustomerName, Phone, Amount and Spouse will solve these issues.

Table 2.1 Child table

ChildID	ChildName
C1001	Philip
C1002	Megan
C1003	Julie
C1004	Pierce
C1005	Johnny

Table 2.2 Customer Table

CustomerID	CustomerName	Phone	Amount	Spouse
A2001	John Bingo	(330) 528-6273	\$5,000	Jenny
A2002	Philip Cusack	(330) 672-5432	\$7,500	
A2004	Charles Dominic	(330) 654-0980	\$25,000	
A2005	Chick Eduardo	(440) 752-6542	\$150,000	Bella
A2110	Will Hanks	(818) 223-7809	\$2,200,000	Lily
A2117	Bruce Willis	(828) 457-1234	\$990,000	
A2120	Remo Williams	(425) 217-5473	\$750,000	
B1201	John Wit	(312) 765-9087	\$450,000	Lisa
B1221	Tom Zack	(330) 298-1891	\$550,000	

Step 2: Create Relationships Between Relations

The primary key of the child table is ChildID and also CustomerID in the customer table. Because a customer has more than one child that makes it a many to one relationship therefore putting the primary key of the customer table in the child table show the relationship between the two themes.

Table 3.2 Child table showing CustomerID.

ChildID	ChildName	CustomerID
C1001	Philip	A2001
C1002	Megan	A2001
C1003	Julie	A2110
C1004	Pierce	B1201
C1005	Johnny	B1201

3. Steps in the Database Development Process

The three major steps involved in database development process are;

- i. **Requirement analysis stage** – During this step, the database designers have to interview the customers (database users) to understand the proposed system and obtain and document the data and functional requirements. The result of this step is a document that includes the detailed requirements provided by the users.

Users requirement are obtained from interviews, reports, queries, application programs, web sites, use cases and business rules. These requirements are used to create data model which is the representation of the content, relationships and constraints on the data needed to support the system requirements. Prototypes or working demonstrations of selected portions of the future system are created during this stage and they are used to obtain feedback from the system users.
- ii. **System design stage** – in this stage, the data model is transformed into a database design. The design includes tables (table names, and column names), relationships (description of primary and foreign keys) and constraints (entity relationship constraint and referential integrity constraints).
- iii. **Implementation stage** – this step involves constructing the database management system, creating the development applications, testing the system, developing operational procedure and documentation, training the users and populating the databases in the same data file of the same organization. It is an important step in database development process.

4.

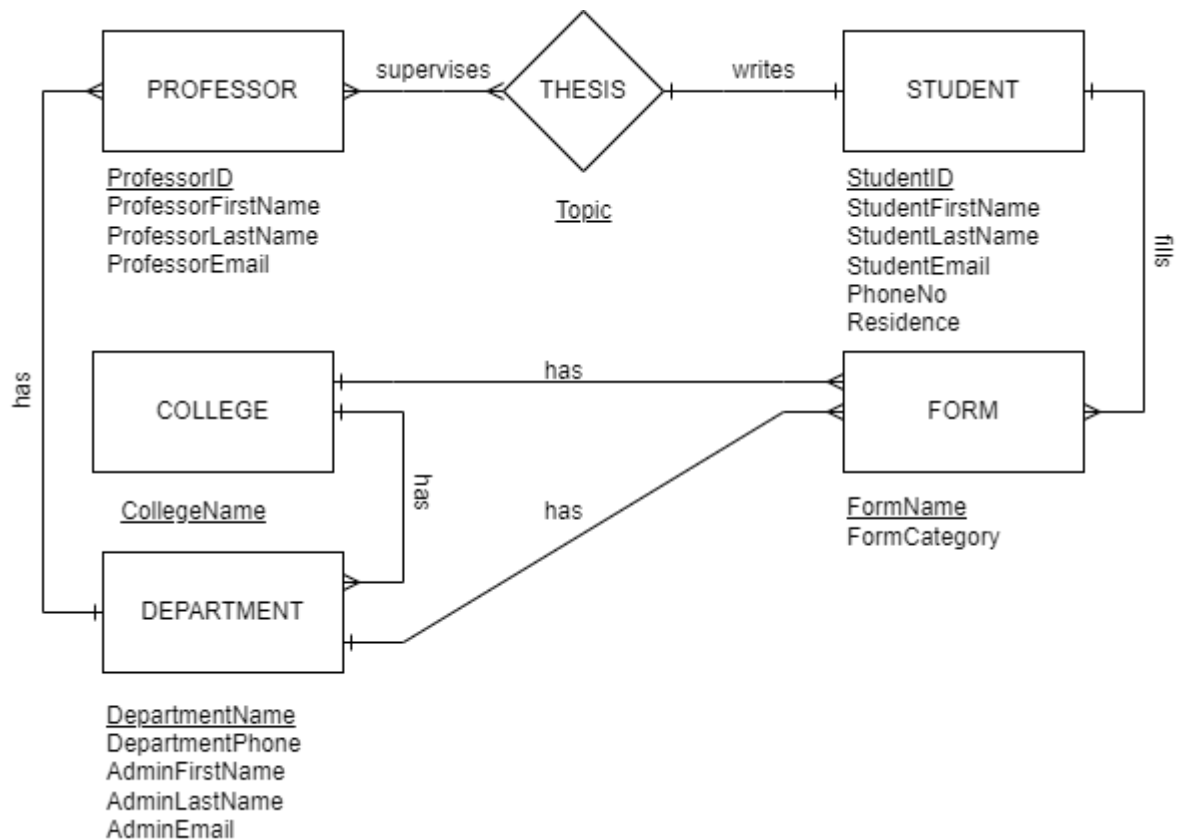


Fig. 4.1 Tree Top University ERD.

I created six entities which are professor, student, thesis, form, college and department. The professor entity has professorID as the primary key because other attributes like the ProfessorFirstName, ProfessorLastName, and ProfessorEmail can be repeated if the two professors have the first and last name. The student entity attributes are StudentID (primary key) StudentFirstName, StudentLastName, StudentEmail, PhoneNo and Residence. The thesis entity is an associative entity because it links the student and professor entities and holds attributes specific to the relationship and the these topic is the primary key. The form entity has FormName has its primary key for example GRE and FormCategory like test scores. The college primary key is CollegeName and the department entity has attributes which are DepartmentName (primary key), DepartmentPhone, AdminFirstName, AdminLastName and AdminEmail.

The assumptions I made for the diagram above are;

- i. A graduate student can only write one thesis and a single thesis is written by one student so it is a one to one relationship. Also, a professor is assigned to supervise many student theses and a thesis is supervised by many professors which makes it a many to many relationship.
- ii. The student works on one thesis and has two or more professors that supervise the thesis work. This makes thesis an associative entity.
- iii. A form can only be filled by a student but a student can fill many forms so it is a one to many relationship.
- iv. A department only belongs to one college but a college has many departments that makes it a one to many relationship.
- v. A professor belongs to only one department but a department has many professors so it is a one to many relationship.
- vi. A single form filled by the student that is submitted to the college administration office is a one to one relationship. The college has many student application forms like application forms which makes it a one to many relationship.
- vii. A department has many forms filled by its student but these forms belong only to that department and other departments can only have access to it when it is requested so the relationship between the department and form is a one to many relationship.