# 2. RELATION SCHEMAS

## 2.1 Faculty

**Relational Model:**

Faculty(facultyName)

**Functional Dependencies:**

No functional dependency.

**Candidate Keys:**

{ (facultyName) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE faculty(

facultyName char(30) PRIMARY KEY

)ENGINE = InnoDB;

## 2.2 Department

**Relational Model:**

Department(deptName, facultyName)

**Functional Dependencies:**

deptName -> facultyName

**Candidate Keys:**

{ (deptName) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE department(

deptName char(5) PRIMARY KEY,

facultyName char(30),

FOREIGN KEY (facultyName) references faculty

)ENGINE = InnoDB;

## 2.3 Person

**Relational Model:**

Person (userID, name, surname, password, phone, deptName)

**Functional Dependencies:**

userID -> name, surname, password, phone, deptName

**Candidate Keys:**

{ (userID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE person(

userID int PRIMARY KEY AUTO\_INCREMENT,

name char(100) NOT NULL,

surname char(100) NOT NULL

password char(12) NOT NULL,

address char(100),

phone char(15),

deptName char(5),

FOREIGN KEY (deptName) references department

)ENGINE = InnoDB;

## 2.4 Student

**Relational Model:**

Student (userID, cgpa)

**Functional Dependencies:**

userID -> cgpa

**Candidate Keys:**

{ (userID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE student(

userID int PRIMARY KEY,

cgpa float,

FOREIGN KEY (userID) references person

)ENGINE = InnoDB;

## 2.5 StudentAdvisor

**Relational Model:**

Student (userID)

**Functional Dependencies:**

No functional dependency.

**Candidate Keys:**

{ (userID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE studentAdvisor(

userID int PRIMARY KEY,

FOREIGN KEY (userID) references person

)ENGINE = InnoDB;

## 2.6 Secretary

**Relational Model:**

Student (userID)

**Functional Dependencies:**

No functional dependency.

**Candidate Keys:**

{ (userID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE secretary(

userID int PRIMARY KEY,

FOREIGN KEY (userID) references person

)ENGINE = InnoDB;

## 2.7 Company

**Relational Model:**

Company (compID, name, password, address, phone, applicableDepts, approvelStatus, supervisorName, supervisorPhone, city, country, evaluatorRating, studentRating, sector)

**Functional Dependencies:**

compID -> name, password, address, phone, applicableDepts, approvelStatus, supervisorName, supervisorPhone, city, country, evaluatorRating, studentRating, sector)

**Candidate Keys:**

{ (compID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE company(

compID int PRIMARY KEY AUTO\_INCREMENT,

name char(100) NOT NULL,

password char(12) NOT NULL,

address char(100),

phone char(15),

applicableDepts char(30),

status char(15) NOT NULL,

supervisorName char(30),

supervisorPhone char(15),

city char(20),

country char(20),

evaluatorRating int,

studentRating int,

sector char(20)

)ENGINE = InnoDB;

## 2.8 RegisteredCompany

**Relational Model:**

RegisteredCompany (compID,registrationDate, studentID)

**Functional Dependencies:**

compID -> registrationDate, studentID

**Candidate Keys:**

{ (compID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE registeredCompany(

compID int PRIMARY KEY,

registrationDate TIMESTAMP,

FOREIGN KEY (compID) references company

FOREIGN KEY (studentID) references student

)ENGINE = InnoDB;

## 2.9 SelfFoundCompany

**Relational Model:**

SelfFoundCompany (compID, advisorID)

**Functional Dependencies:**

compID -> advisorID

**Candidate Keys:**

{ (compID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE selfFoundCompany (

compID int PRIMARY KEY,

FOREIGN KEY (compID) references company

FOREIGN KEY (advisorID) references studentAdvisor

)ENGINE = InnoDB;

## 2.10 Quota

**Relational Model:**

Quota (quotaID, internshipDuration, internshipStartDate, internshipEndDate, availableYears, status, quotaAmount, quotaDeadline,compID, generalAnnouncementID)

**Functional Dependencies:**

quotaID -> internshipDuration, internshipStartDate, internshipEndDate,availableYears, status,

quotaAmount, quotaDeadline, compID

**Candidate Keys:**

{ (quotaID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE quota(

quotaID int PRIMARY KEY AUTO\_INCREMENT,

internshipDuration int NOT NULL,

internshipStartDate date NOT NULL,

internshipEndDate date NOT NULL,

availableYears int NOT NULL,

status char(30),

quotaAmount int NOT NULL,

quotaDeadline date NOT NULL,

FOREIGN KEY (generalAnnouncementID) references generalAnnouncement,

)ENGINE = InnoDB;

**2.11 Opens**

**Relational Model:**

Opens(quotaID, compID, deptName)

**Functional Dependencies:**

No functional dependency.

**Candidate Keys:**

{ (quotaID, compID, deptName) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE opens (

quotaID int,

compID int,

deptName char(5),

FOREIGN KEY (quotaID) references quota,

FOREIGN KEY (compID) references registeredCompany,

FOREIGN KEY (deptName) references department

)ENGINE = InnoDB;

## 2.12 Application

**Relational Model:**

Application (appID, appSubmitDate, approval, secretaryID)

**Functional Dependencies:**

appID ->appSubmitDate, approval, secretaryID

**Candidate Keys:**

{ (appID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE application(

appID int PRIMARY KEY AUTO\_INCREMENT,

appSubmitDate TIMESTAMP,

approval char(12),

secretaryID int,

FOREIGN KEY (secretaryID) references secretary

)ENGINE = InnoDB;

## 2.13 QuotaApply

**Relational Model:**

QuotaApply(appID, quotaID, compID, studentID, drawResult)

**Functional Dependencies:**

appID ->quotaID, compID, studentID, drawResult

**Candidate Keys:**

{ (appID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE quotaApply(

appID int PRIMARY KEY,

quotaID int,

compID int,

studentID int,

drawResult int,

FOREIGN KEY (appID) references application,

FOREIGN KEY (quotaID) references quota,

FOREIGN KEY (compID) references registeredCompany,

FOREIGN KEY (studentID) references student

)ENGINE = InnoDB;

## 2.14 DirectApply

**Relational Model:**

DirectApply(appID, compID, studentID)

**Functional Dependencies:**

appID -> compID, studentID

**Candidate Keys:**

{ (appID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE directApply(

appID int PRIMARY KEY,

compID int,

studentID int,

FOREIGN KEY (appID) references application,

FOREIGN KEY (compID) references registeredCompany,

FOREIGN KEY (studentID) references student

)ENGINE = InnoDB;

## 2.15 Announcement

**Relational Model:**

Announcement(announcementID, date)

**Functional Dependencies:**

announcementID -> date

**Candidate Keys:**

{ (secretaryID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE announcement(

announcementID int PRIMARY KEY AUTO\_INCREMENT,

date TIMESTAMP

)ENGINE = InnoDB;

## 2.16 AppFeedbackAnnouncement

**Relational Model:**

AppFeedbackAnnouncement(announcementID, studentApproval, deadline, studentID, secretaryID)

**Functional Dependencies:**

announcementID -> studentApproval, deadline, secretaryID

**Candidate Keys:**

{ (announcementID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE appFeedbackAnnouncement (

announcementID int PRIMARY KEY,

studentApproval char(3),

deadline date,

secretaryID int,

FOREIGN KEY (announcementID) references announcement,

FOREIGN KEY (secretaryID) references secretary,

FOREIGN KEY (studentID) references student

)ENGINE = InnoDB;

## 2.17 GeneralAnnouncement

**Relational Model:**

GeneralAnnouncement(announcementID, title, message, secretaryID)

**Functional Dependencies:**

announcementID -> title, message, secretaryID

**Candidate Keys:**

{ (announcementID) }

**Normal Form:**

BCNF

**Table Definition:**

CREATE TABLE generalAnnouncement (

announcementID int PRIMARY KEY,

title char(30),

message varchar(7000),

secretaryID int,

FOREIGN KEY (announcementID) references announcement,

FOREIGN KEY (secretaryID) references secretary

)ENGINE = InnoDB;