University of Dhaka

Department of Computer Science & Engineering University of Dhaka

Database System Lab Assignment

Course Code: CSE-2211

2nd year 2nd semester 2016

A School Database Model

## Submitted To:

Mr. Abu Ahmed Ferdaus

Associate Professor

Department of Computer Science & Engineering

Submitted By:

Md. Aminul Islam

Roll: AE-122

Batch-21

2nd year 2nd semester, 2016

Department of Computer Science Engineering

### Submission Date:

29-NOV-2016

Index

Content Page No

1. Introduction………………………………………………………………………………………………………..4

2. Schemas with attribute……………………………………………………………………………..………….4

3.DDL and Snapshots………………………………………………………………………………………..……..5

3.1. Classroom (Room\_Number, Capacity)………………………………………………….….5

3.2. Class (Class\_Name, Seat, Room\_Number)…………………………………………………6

3.3. Subject (Sub\_ID, Title, Class\_Name)………………………………………………………….8

3.4. Teacher (ID, Name, Salary)……………………………………………………………………….9

3.5. Section (Sub\_ID, Sec\_ID, Year, Room\_Number, Routine)…………………….……..11

3.6. Teaches (ID, Sub\_ID, Sec\_ID, Year)……………………………………………………..…….12

3.7. Student (ID, Name, Class\_Name, DOB)………………………………………………………14

3.8. Routine (Routine\_ID, Day, Start\_Hr, End\_Hr)…………………………………………….15

4. Schema Diagram……………………………………………………………………………………………….…..17

5. Query and SQL Statements…………………………………………………………………………….……. 18

5.1 Natural join…………………………………………………………………………………..………….18

5.2 Cross Product………………………………………………………………………………………….18

5.3 Outer join……………………………………………………………………………………….……….19

5.4 Join (with using on)………………………………………………………………………..……….19

5.5 Grouping and Ordering……………………………………………………………………………..21

5.6 Having clauses……………………………………….…………………………………………………22

5.7 Nested Sub Query(some, all, exists, in)…………………………..…………………………...23

5.8 Aggregation(count, avg, max, min, sum)……………………………….…………………….25

5.9 Set Operation(union, intersect, minus)………………………………………………………28

5.10 Modification(insert, delete, update)…………………………………………………………30

6. Desired Normal Forms in the College Database……………………………………………………….32

6.1 First Normal Form (1NF)……………………………………………………………………………32

6.2 Boyce–Codd Normal Form (BCNF)……………………………………………………………..32

6.3 Third Normal Form (3NF)………………………………………………………………………….33

7. Non-trivial Functional Dependencies………………………………………………………………………34

7.1. Classroom (Room\_Number, Capacity) ……………………………………………………….34

7.2. Class (Class\_Name, Seat, Room\_Number)…………………………………………………..34

7.3. Subject (Sub\_ID, Title, Class\_Name)……………………………………………………………34

7.4. Teacher (ID, Name, Salary)………………………………………………………………………...34

7.5. Section (Sub\_ID, Sec\_ID, Year, Room\_Number, Routine)………………………………35

7.6. Teaches (ID, Sub\_ID, Sec\_ID, Year)………………………………………………………………35

7.7. Student (ID, Name, Class\_Name, DOB)…………………………………………………………35

7.8. Routine (Routine\_ID, Day, Start\_Hr, End\_Hr)………………………………………………..36

8. Discussions………………………………………………………………………………………………………………36

A School Database

# 1. Introduction:

A database system project is about to produce of our own and ask some different queries on that database. I decided to make a database of a school. In that database, I have made 8 relational table to accomplish the database. The tables are-

‘Classroom’, ‘Class’, ‘Subject’, ‘Teacher’, ‘Section’, ‘Teaches’, ‘Student’ and ‘Routine’.

There are 3 independent table called ‘Classroom’ , ‘Teacher’ and ‘Routine’. The information of class room, teachers and routine are kept in this three tables respectively. Another information are kept in other 6 tables which are related with independent tables. ‘Subject’ and ‘Student’ relation are depend on the ‘Class’ relation where ‘Class’ relation is dependent on independent ‘Classroom’ relation. ‘Section’ depends on ‘Subject’ and ‘Classroom’ and ‘Teaches’ relation depends on ‘Teacher’ and ‘Subject’ relation. In the relation I insert some information that are collected from a local school. With that relation and information I tried to make a database system and make some query with the answers.

# 2. Schemas with attribute:

I have mentioned earlier that I made 8 schemas this schemas have several attribute. The schemas and attributes are-

1. Classroom (Room\_Number, Capacity)

2. Class (Class\_Name, Seat, Room\_Number)

3. Subject (Sub\_ID, Title, Class\_Name)

4. Teacher (ID, Name, Salary)

5. Section (Sub\_ID, Sec\_ID, Year, Room\_Number, Routine)

6. Teaches (ID, Sub\_ID, Sec\_ID, Year)

7. Student (ID, Name, Class\_Name, DOB)

8. Routine (Routine\_ID, Day, Start\_Hr, End\_Hr)

# 3.DDL and Snapshots:

**3.1. Classroom (Room\_Number, Capacity)**

*CREATE TABLE "CLASSROOM"*

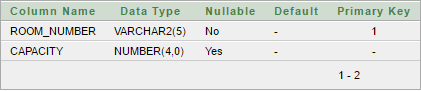
*( "ROOM\_NUMBER" VARCHAR2(5),*

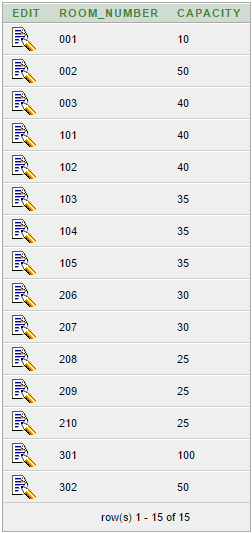
*"CAPACITY" NUMBER(4,0),*

*CONSTRAINT "CLSRM\_RM\_NMR\_PK" PRIMARY KEY ("ROOM\_NUMBER") ENABLE*

*)*

*/*





**3.2. Class (Class\_Name, Seat, Room\_Number)**

*CREATE TABLE "CLASS"*

*( "CLASS\_NAME" VARCHAR2(20),*

*"SEAT" NUMBER(3,0),*

*"ROOM\_NUMBER" VARCHAR2(5),*

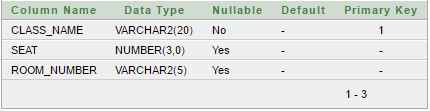
*CONSTRAINT "CLS\_CLS\_NM\_PK" PRIMARY KEY ("CLASS\_NAME") ENABLE,*

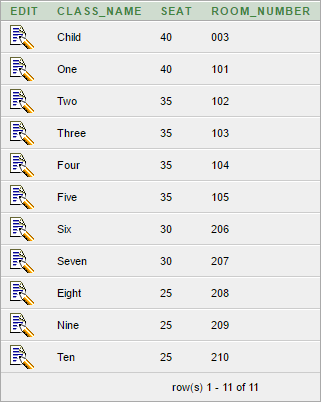
*CONSTRAINT "CLS\_RM\_NMR\_FK" FOREIGN KEY ("ROOM\_NUMBER")*

*REFERENCES "CLASSROOM" ("ROOM\_NUMBER") ON DELETE SET NULL ENABLE*

*)*

*/*





**3.3. Subject (Sub\_ID, Title, Class\_Name)**

*CREATE TABLE "SUBJECT"*

*( "SUB\_ID" VARCHAR2(10),*

*"TITLE" VARCHAR2(50),*

*"CLASS\_NAME" VARCHAR2(20),*

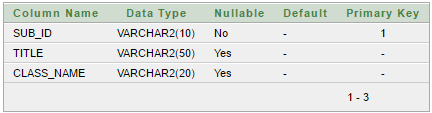
*CONSTRAINT "SUB\_SUB\_ID\_PK" PRIMARY KEY ("SUB\_ID") ENABLE,*

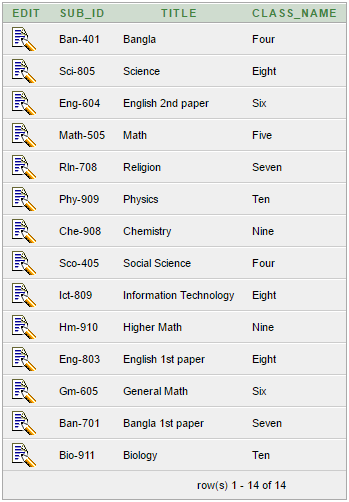
*CONSTRAINT "SUB\_CLS\_NM\_FK" FOREIGN KEY ("CLASS\_NAME")*

*REFERENCES "CLASS" ("CLASS\_NAME") ON DELETE SET NULL ENABLE*

*)*

*/*

**

**

**3.4. Teacher (ID, Name, Salary)**

*CREATE TABLE "TEACHER"*

*( "ID" VARCHAR2(6),*

*"NAME" VARCHAR2(20),*

*"SALARY" NUMBER(8,2),*

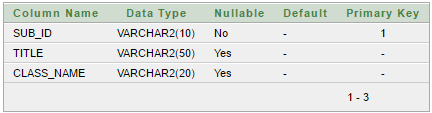
*CONSTRAINT "TCHR\_ID\_PK" PRIMARY KEY ("ID") ENABLE,*

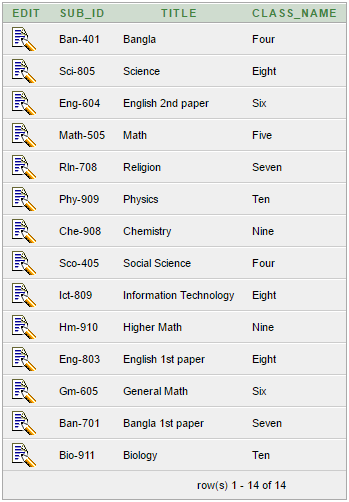
*CONSTRAINT "TCHR\_NM\_CHK1" CHECK (name is not null) ENABLE,*

*CONSTRAINT "TCHR\_SAL\_CHK2" CHECK (salary > 15000) ENABLE*

*)*

*/*

**

**

**3.5. Section (Sub\_ID, Sec\_ID, Year, Room\_Number, Routine)**

*CREATE TABLE "SECTION"*

*( "SUB\_ID" VARCHAR2(10),*

*"SEC\_ID" VARCHAR2(8),*

*"YEAR" NUMBER(4,0),*

*"ROOM\_NUMBER" VARCHAR2(5),*

*"ROUTINE\_ID" VARCHAR2(4),*

*CHECK (year > 1701 and year < 2100) ENABLE,*

*PRIMARY KEY ("SUB\_ID", "SEC\_ID", "YEAR") ENABLE,*

*FOREIGN KEY ("SUB\_ID")*

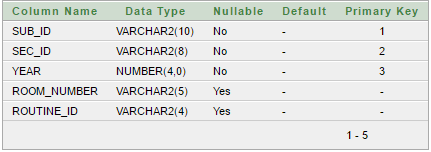
*REFERENCES "SUBJECT" ("SUB\_ID") ON DELETE CASCADE ENABLE,*

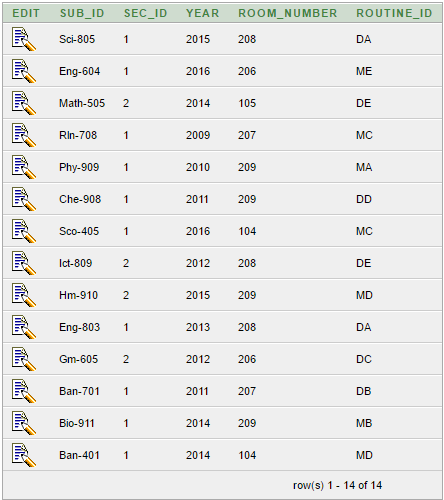
*FOREIGN KEY ("ROOM\_NUMBER")*

*REFERENCES "CLASSROOM" ("ROOM\_NUMBER") ON DELETE SET NULL ENABLE*

*)*

*/*

**

**

**3.6. Teaches (ID, Sub\_ID, Sec\_ID, Year)**

*CREATE TABLE "TEACHES"*

*( "ID" VARCHAR2(6),*

*"SUB\_ID" VARCHAR2(10),*

*"SEC\_ID" VARCHAR2(8),*

*"YEAR" NUMBER(4,0),*

*PRIMARY KEY ("ID", "SUB\_ID", "SEC\_ID", "YEAR") ENABLE,*

*FOREIGN KEY ("SUB\_ID", "SEC\_ID", "YEAR")*

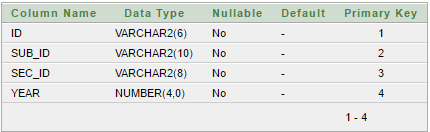
*REFERENCES "SECTION" ("SUB\_ID", "SEC\_ID", "YEAR") ON DELETE CASCADE ENABLE,*

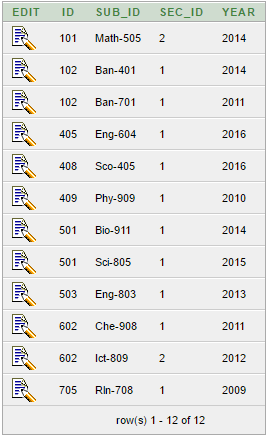
*FOREIGN KEY ("ID")*

*REFERENCES "TEACHER" ("ID") ON DELETE CASCADE ENABLE*

*)*

*/*

**

**

**3.7. Student (ID, Name, Class\_Name, DOB)**

*CREATE TABLE "STUDENT"*

*( "ID" VARCHAR2(6),*

*"NAME" VARCHAR2(50),*

*"CLASS\_NAME" VARCHAR2(20),*

*"DOB" DATE,*

*CONSTRAINT "STD\_ID\_PK" PRIMARY KEY ("ID") ENABLE,*

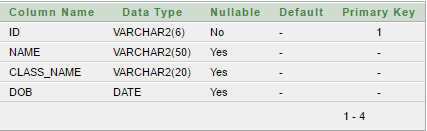
*CONSTRAINT "STD\_NM\_CHK" CHECK (name is not null) ENABLE,*

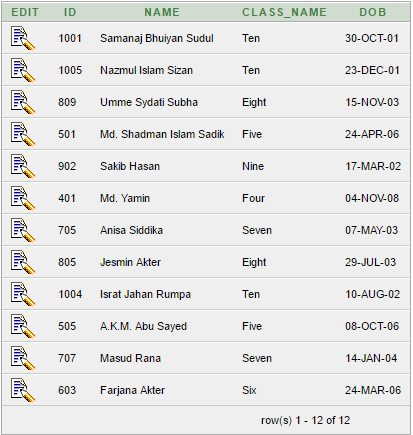
*CONSTRAINT "STD\_CLS\_NM\_FK" FOREIGN KEY ("CLASS\_NAME")*

*REFERENCES "CLASS" ("CLASS\_NAME") ON DELETE SET NULL ENABLE*

*)*

*/*

**

**

**3.8. Routine (Routine\_ID, Day, Start\_Hr, End\_Hr)**

*CREATE TABLE "ROUTINE"*

*( "ROUTINE\_ID" VARCHAR2(4),*

*"DAY" VARCHAR2(10),*

*"START\_HR" NUMBER(2,0),*

*"END\_HR" NUMBER(2,0),*

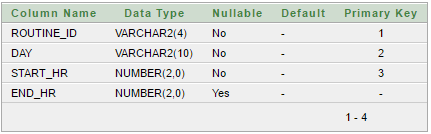
*CHECK (start\_hr >= 0 and start\_hr < 24) ENABLE,*

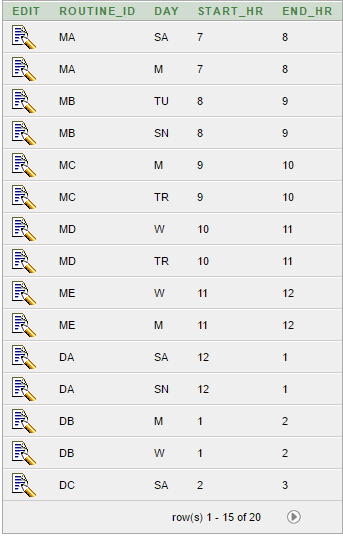
*CHECK (end\_hr >= 0 and end\_hr < 24) ENABLE,*

*PRIMARY KEY ("ROUTINE\_ID", "DAY", "START\_HR") ENABLE*

*)*

*/*

**

**

# 4. Schema Diagram:

# E:\Academic\4th semester\CSE-2201-Database\Lab\School Database\SchemaDiagram.png

# 5. Query and SQL Statements:

A query language is a language in which a user requests information from the database.

**5.1 Natural join:**

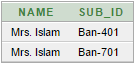
Query:

For listing the name of the teacher along with the subject ID that he/she taught where id is ‘101’.

SQL:

*select name, sub\_id  
from teacher natural join teaches*

*where id='102'*



**5.2 Cross Product:**

Query:

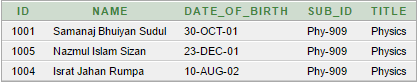
For listing the id, names, Date of birth, title of subject of the students who the the subject ‘Phy-909’.

SQL:

select id, name, dob as Date\_of\_Birth, title

from student,subject

where student.class\_name=subject.class\_name and sub\_id='Phy-909'



**5.3 Outer join:**

Query:

Listing the name, class name, date of birth subject id and subject title for the student id ‘707’

SQL:

select name,class\_name,dob as date\_of\_birth,sub\_id,title

from student natural left outer join subject

where id ='707'

D:\Lab\School Database\3.PNG

**5.4 Join (with using on):**

Using:

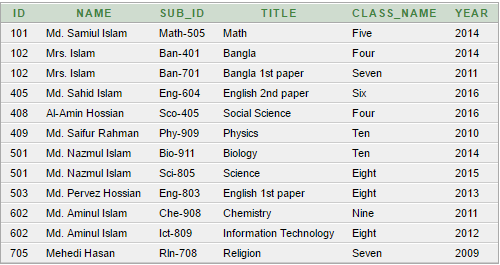
Query:

Finding the id, name of a teacher who taught different subject in different year along with sub\_id, title, year and class name

SQL:

*select id,name,sub\_id,title,class\_name,year*

*from (teacher natural join teaches)join subject using (sub\_id)*



With:

Query:

Finding the id and name who get the maximum salary in the school.

SQL:

with max\_salary(sal) as

(select max(salary)

from teacher)

select id,name

from teacher,max\_salary

where teacher.salary=max\_salary.sal

D:\Lab\School Database\5.PNG

On:

Query:

Finding the room number, capacity, number of seat for the class ‘Child’.

SQL:

select \*

from classroom join class on classroom.room\_number=class.room\_number

where class\_name='Child'

D:\Lab\School Database\6.PNG

**5.5 Grouping and Ordering:**

Grouping:

Query:

Finding the number of teacher in different years who taught in section 1 grouping by year.

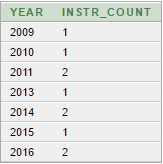
SQL:

*select year, count (distinct ID) as instr\_count*

*from teacher natural join teaches*

*where sec\_id='1'*

*group by year;*



Ordering:

Query:

Finding the id, name and date of birth of student of class eight ordering by their name.

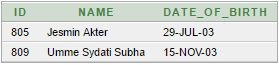
SQL:

*select id,name,dob as Date\_of\_birth*

*from student*

*where class\_name='Eight'*

*order by name*



**5.6 Having clauses:**

Query:

Finding the number of teacher in different years who taught in section 1 where number of teacher is greater than 1.

SQL:

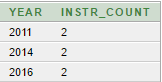
*select year, count (distinct ID) as instr\_count*

*from teacher natural join teaches*

*where sec\_id='1'*

*group by year*

*having count(id)>1*



**5.7 Nested Sub Query(some, all, exists, in):**

Some:

Query:

Finding any teacher that his/her salary is greater than 25000.

SQL:

*select name*

*from teacher*

*where salary> some(select salary*

*from teacher*

*where salary>25000)*

D:\Lab\School Database\10.PNG

All:

Query:

Finding all teachers that his/her salary is less than 25000.

SQL:

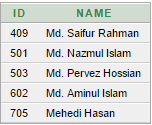
*select id,name*

*from teacher*

*where salary< all(select salary*

*from teacher*

*where salary>=25000)*



Exists:

Query:

Finding the subject id that is taught in 2012 and 2010.

SQL:

*select sub\_id*

*from section as s*

*where year=2012 and exists(select \**

*from section t*

*where year=2010 and*

*s.sub\_id=t.sub\_id)*

No data found for this query.

In:

Query:

Finding the number of student id who is in class ten.

SQL:

*select count (distinct ID)*

*from student*

*where (id,name) in   
(select id,name*

*from student*

*where student.class\_name= 'Ten')*

D:\Lab\School Database\13.PNG

**5.8 Aggregation(count, avg, max, min, sum):**

Count:

Query:

Finding the distinct id who taught in 2014.

SQL:

*select count (distinct ID) as no\_of\_id*

*from teaches*

*where year = 2014*



Avg:

Query:

Finding the average salary of the teachers

SQL:

*select avg(salary) as avg\_sal*

*from teacher*



Max:

Query:

Finding the maximum capacity in a room in the whole school.

*select max(capacity)*

*from classroom*

D:\Lab\School Database\16.PNG

Min:

Query:

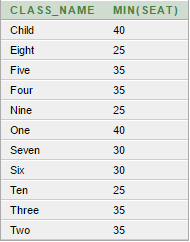
Finding the minimum number of seat of every class.

SQL:

*select class\_name,min(seat)*

*from class*

*group by class\_name*



Sum:

Query:

Total number of seat in the school.

SQL:

*select sum(seat)*

*from class*

D:\Lab\School Database\18.PNG

**5.9 Set Operation(union, intersect, minus):**

Union:

Query:

Finding the subject id/ids which are taught in 2014 or 2016.

SQL:

*select sub\_id*

*from section*

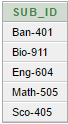
*where year = 2014*

*union*

*(select sub\_id*

*from section*

*where year = 2016)*



Intersect:

Query:

Finding the subject id/ids which are taught in both 2010 and 2013.

SQL:

*select sub\_id*

*from section*

*where year = 2010*

*intersect*

*(select sub\_id*

*from section*

*where year = 2013)*

No data found for this query.

Minus:

Query:

Finding the subject id/ids which are taught in 2014 but not in 2011

SQL:

*select sub\_id*

*from section*

*where year = 2014*

*minus*

*(select sub\_id*

*from section*

*where year = 2011)*



**5.10 Modification(insert, delete, update):**

Insertion:

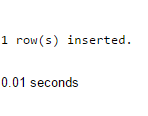
Query:

Inserting a students information.

SQL:

*insert into student*

*values ('603', 'Farjana Akter', 'Six','24-Mar-06')*



Update:

Query:

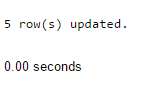
Update the salary of teachers who have salary less than 25000 with 10%.

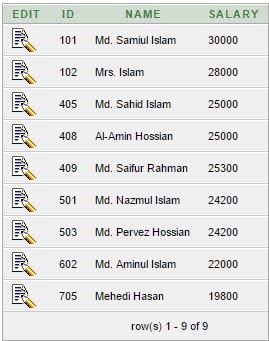
SQL:

*update teacher*

*set salary = salary \* 1.1*

*where salary <25000;*





Delete:

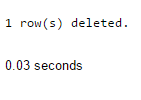
Query:

Deleting a row from teacher table.

SQL:

*delete from teacher*

*where salary<20000*



**6. Desired Normal Forms in the College Database**

**6.1 First Normal Form (1NF):**

A domain is atomic if elements of the domain are considered to be indivisible units. We say that a relation schema *R* is in First Normal Form (1NF) if the domains of all attributes of *R* are atomic. In the school database domains of all attributes are not atomic. So we can say that the school database is not in First Normal Form (1NF).

**6.2 Boyce–Codd Normal Form (BCNF):**

A relation schema R is in BCNF with respect to a set F of functional dependencies if, for all functional dependencies in F+ of the form A → B, where A ⊆ R and B ⊆ R, at least one of the following holds:

• A → B is a trivial functional dependency (that is, B ⊆ A).

• A is a super key for schema R.

A database design is in BCNF if each member of the set of relation schemas that constitutes the design is in BCNF.

• In the classroom relation (Room\_Number, Capacity) is a super key.

• In Class relation Class\_Name is a primary key.

• In Subject relation(Sub\_ID, Title) is super key.

• In Teacher relation (ID, Name) is a super key.

• In Section relation (Sub\_ID, Sec\_ID, Year) is a primary key.

• In Teaches relation (ID, Sub\_ID, Sec\_ID, Year) is a primary key.

• In Student relation ID is a primary key.

• In Routine relation (Routine\_ID, Day, Start\_Hr) is a super key.

We see from the above functional dependency sets that all our schemas hold the BCNF.

**6.3 Third Normal Form (3NF):**

A relation schema *R* is in Third Normal Form with respect to a set *F* of functional dependencies if, for all functional dependencies in *F*+ of the form A→ B, where A ⊆ R and B ⊆ R, at least one of the following holds:

• A → B is a trivial functional dependency (that is, B ⊆ A).

• A is a super key for schema R.

• Each attribute C in B− A is contained in a candidate key for R.

We see that if a database holds BCNF then it automatically holds 3NF as one of the above mentioned conditions for 3NF are met because at least 1 of the first 2 conditions must met to complete the BCNF. As the school database holds BCNF , the school database also holds the 3NF.

**7. Non-trivial Functional Dependencies**

In school database management system there are 8 relations and all of them have many functional dependencies among their own attributes. Given a relational schema r (R), a functional dependency f on R is logically implied by a set of functional dependencies F on r if every instance of r (R) that satisfies F also satisfies f .

Some non-trivial functional dependencies are listed below –

**7.1. Classroom (Room\_Number, Capacity)**

Room\_Number🡪 Capacity

**7.2. Class (Class\_Name, Seat, Room\_Number)**

Class\_Name 🡪 Room\_Number

Class\_Name 🡪 Seat

Class\_Name🡪 Seat, Room\_Number

Class\_Name, Seat 🡪 Room\_Number

Class\_Name, Room\_Number 🡪 Seat

**7.3. Subject (Sub\_ID, Title, Class\_Name)**

Sub\_ID 🡪 Title

Sub\_ID 🡪 Class\_Name

Sub\_ID 🡪 Title, Class\_Name

Sub\_ID, Title 🡪 Class\_Name

Sub\_ID, Class\_Name 🡪 Title

**7.4. Teacher (ID, Name, Salary)**

ID 🡪 Salary

ID 🡪 Name,

ID 🡪 Name, Salary

ID, Name 🡪 Salary

ID, Salary 🡪 Name

**7.5. Section (Sub\_ID, Sec\_ID, Year, Room\_Number, Routine)**

Sub\_ID, Sec\_ID, Year 🡪 Room\_Number

Sub\_ID, Sec\_ID, Year 🡪 Routine

Sub\_ID, Sec\_ID, Year 🡪 Room\_Number, Routine

Sub\_ID, Sec\_ID, Year, Room\_Number 🡪 Routine

Sub\_ID, Sec\_ID, Year, Routine 🡪 Room\_Number

**7.6. Teaches (ID, Sub\_ID, Sec\_ID, Year)**

There is no non-trivial functional dependency in this Teaches relation.

**7.7. Student (ID, Name, Class\_Name, DOB)**

ID 🡪 Name

ID 🡪 Class\_Name

ID 🡪 DOB

ID 🡪Name, Class\_Name

ID 🡪Name, DOB

ID 🡪 Class\_Name, DOB

ID 🡪 Name, Class\_Name, DOB

ID, Name 🡪 Class\_Name

ID, Name 🡪DOB

ID, Name 🡪 Class\_Name, DOB

ID, Class\_Name 🡪 DOB

ID, Class\_Name 🡪 Name

ID,Class\_Name 🡪 Name DOB

ID, DOB🡪 Name

ID, DOB🡪 Class\_Name

ID, DOB🡪 Name, Class\_Name

ID, Name, Class\_Name 🡪DOB

ID, Name, DOB 🡪Class\_Name

ID, Class\_Name, DOB 🡪 Name

**7.8. Routine (Routine\_ID, Day, Start\_Hr, End\_Hr)**

Routine\_ID, Day, Start\_Hr 🡪 End\_Hr

**8. Discussions:**

After doing this assignment I have learnt many things about database management system. I also face some difficulties while doing this assignment. I have learnt how to create a relation, how to insert data, specially I have learnt about SQL query language. I have learnt functional dependency, how to draw a schema diagram. I think this assignment will help to understand the database system and this will help in future.