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

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

)

/

Column Name	Data Type	Nullable	Default	Primary Key
ROOM_NUMBER	VARCHAR2(5)	No	-	1
CAPACITY	NUMBER(4,0)	Yes	-	-
				1 - 2

EDIT	ROOM_NUMBER	CAPACITY
	001	10
	002	50
	003	40
	101	40
	102	40
	103	35
	104	35
	105	35
	206	30
	207	30
	208	25
	209	25
	210	25
	301	100
	302	50
	row(s) 1 - 15 of 15	

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

)

/

Column Name	Data Type	Nullable	Default	Primary Key
CLASS_NAME	VARCHAR2(20)	No	-	1
SEAT	NUMBER(3,0)	Yes	-	-
ROOM_NUMBER	VARCHAR2(5)	Yes	-	-
				1 - 3

EDIT	CLASS_NAME	SEAT	ROOM_NUMBER
	Child	40	003
	One	40	101
	Two	35	102
	Three	35	103
	Four	35	104
	Five	35	105
	Six	30	206
	Seven	30	207
	Eight	25	208
	Nine	25	209
	Ten	25	210
	row(s) 1 - 11 of 11		

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

)

/

Column Name	Data Type	Nullable	Default	Primary Key
SUB_ID	VARCHAR2(10)	No	-	1
TITLE	VARCHAR2(50)	Yes	-	-
CLASS_NAME	VARCHAR2(20)	Yes	-	-
				1 - 3

EDIT	SUB_ID	TITLE	CLASS_NAME
	Ban-401	Bangla	Four
	Sci-805	Science	Eight
	Eng-604	English 2nd paper	Six
	Math-505	Math	Five
	Rln-708	Religion	Seven
	Phy-909	Physics	Ten
	Che-908	Chemistry	Nine
	Sco-405	Social Science	Four
	Ict-809	Information Technology	Eight
	Hm-910	Higher Math	Nine
	Eng-803	English 1st paper	Eight
	Gm-605	General Math	Six
	Ban-701	Bangla 1st paper	Seven
	Bio-911	Biology	Ten
			row(s) 1 - 14 of 14

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

)

/

Column Name	Data Type	Nullable	Default	Primary Key
SUB_ID	VARCHAR2(10)	No	-	1
TITLE	VARCHAR2(50)	Yes	-	-
CLASS_NAME	VARCHAR2(20)	Yes	-	-
				1 - 3

EDIT	SUB_ID	TITLE	CLASS_NAME
	Ban-401	Bangla	Four
	Sci-805	Science	Eight
	Eng-604	English 2nd paper	Six
	Math-505	Math	Five
	Rln-708	Religion	Seven
	Phy-909	Physics	Ten
	Che-908	Chemistry	Nine
	Sco-405	Social Science	Four
	Ict-809	Information Technology	Eight
	Hm-910	Higher Math	Nine
	Eng-803	English 1st paper	Eight
	Gm-605	General Math	Six
	Ban-701	Bangla 1st paper	Seven
	Bio-911	Biology	Ten
			row(s) 1 - 14 of 14

3.5. Section (<u>Sub_ID</u>, <u>Sec_ID</u>, <u>Year</u>, 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

)

/

Column Name	Data Type	Nullable	Default	Primary Key
SUB_ID	VARCHAR2(10)	No	-	1
SEC_ID	VARCHAR2(8)	No	-	2
YEAR	NUMBER(4,0)	No	-	3
ROOM_NUMBER	VARCHAR2(5)	Yes	-	-
ROUTINE_ID	VARCHAR2(4)	Yes	-	-
				1 - 5

EDIT	SUB_ID	SEC_ID	YEAR	ROOM_NUMBER	ROUTINE_ID
	Sci-805	1	2015	208	DA
	Eng-604	1	2016	206	ME
	Math-505	2	2014	105	DE
	Rln-708	1	2009	207	MC
	Phy-909	1	2010	209	MA
	Che-908	1	2011	209	DD
	Sco-405	1	2016	104	MC
	Ict-809	2	2012	208	DE
	Hm-910	2	2015	209	MD
	Eng-803	1	2013	208	DA
	Gm-605	2	2012	206	DC
	Ban-701	1	2011	207	DB
	Bio-911	1	2014	209	MB
	Ban-401	1	2014	104	MD

row(s) 1 - 14 of 14

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

)

/

Column Name	Data Type	Nullable	Default	Primary Key
ID	VARCHAR2(6)	No	-	1
SUB_ID	VARCHAR2(10)	No	-	2
SEC_ID	VARCHAR2(8)	No	-	3
YEAR	NUMBER(4,0)	No	-	4
				1 - 4

EDIT	ID	SUB_ID	SEC_ID	YEAR
	101	Math-505	2	2014
	102	Ban-401	1	2014
	102	Ban-701	1	2011
	405	Eng-604	1	2016
	408	Sco-405	1	2016
	409	Phy-909	1	2010
	501	Bio-911	1	2014
	501	Sci-805	1	2015
	503	Eng-803	1	2013
	602	Che-908	1	2011
	602	Ict-809	2	2012
	705	Rln-708	1	2009
		row(s) 1 - 12 of 12		

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
   )
/
```

Column Name	Data Type	Nullable	Default	Primary Key
ID	VARCHAR2(6)	No	-	1
NAME	VARCHAR2(50)	Yes	-	-
CLASS_NAME	VARCHAR2(20)	Yes	-	-
DOB	DATE	Yes	-	-
				1 - 4

EDIT	ID	NAME	CLASS_NAME	DOB
	1001	Samanaj Bhuiyan Sudul	Ten	30-OCT-01
	1005	Nazmul Islam Sizan	Ten	23-DEC-01
	809	Umme Sydati Subha	Eight	15-NOV-03
	501	Md. Shadman Islam Sadik	Five	24-APR-06
	902	Sakib Hasan	Nine	17-MAR-02
	401	Md. Yamin	Four	04-NOV-08
	705	Anisa Siddika	Seven	07-MAY-03
	805	Jesmin Akter	Eight	29-JUL-03
	1004	Israt Jahan Rumpa	Ten	10-AUG-02
	505	A.K.M. Abu Sayed	Five	08-OCT-06
	707	Masud Rana	Seven	14-JAN-04
	603	Farjana Akter	Six	24-MAR-06
				row(s) 1 - 12 of 12

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

)

/

Column Name	Data Type	Nullable	Default	Primary Key
ROUTINE_ID	VARCHAR2(4)	No	-	1
DAY	VARCHAR2(10)	No	-	2
START_HR	NUMBER(2,0)	No	-	3
END_HR	NUMBER(2,0)	Yes	-	-
				1 - 4

EDIT	ROUTINE_ID	DAY	START_HR	END_HR
	MA	SA	7	8
	MA	M	7	8
	MB	TU	8	9
	MB	SN	8	9
	MC	M	9	10
	MC	TR	9	10
	MD	W	10	11
	MD	TR	10	11
	ME	W	11	12
	ME	M	11	12
	DA	SA	12	1
	DA	SN	12	1
	DB	M	1	2
	DB	W	1	2
	DC	SA	2	3
			row(s) 1 - 15 of 20	

4. Schema Diagram:
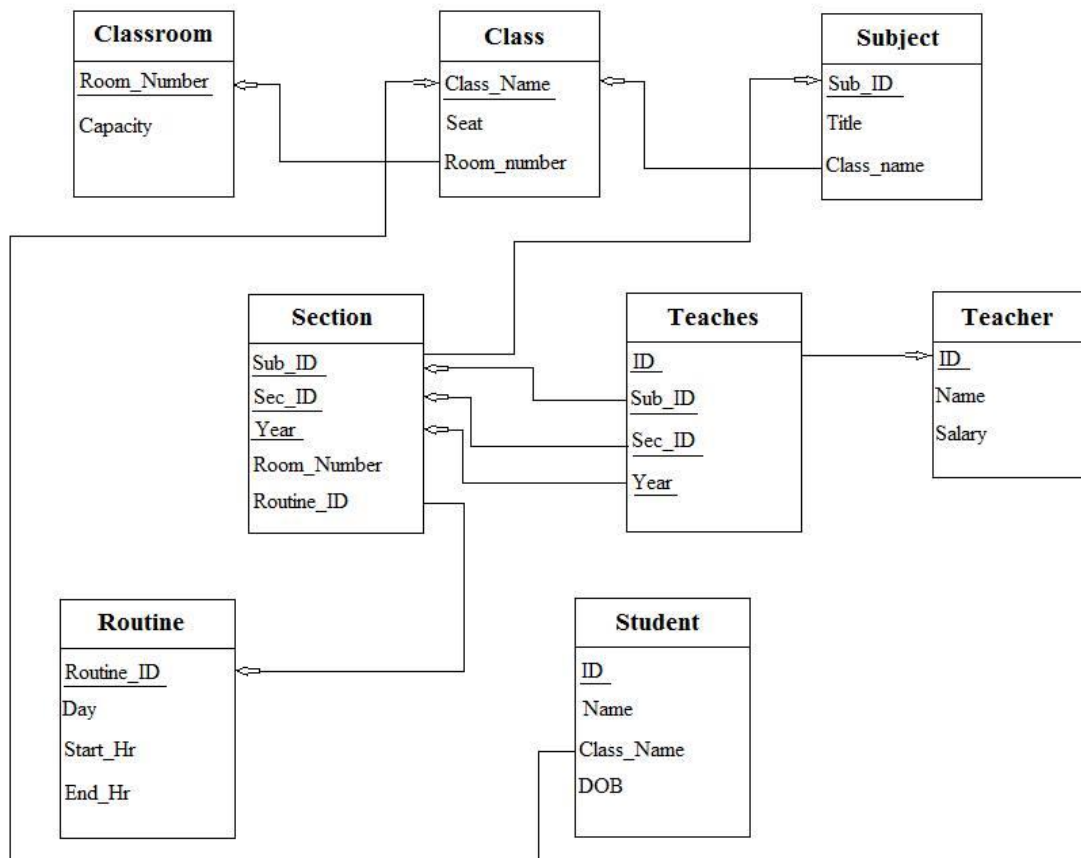
5. Query and SQL Statements:

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'

NAME	SUB_ID
Mrs. Islam	Ban-401
Mrs. Islam	Ban-701

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'

ID	NAME	DATE_OF_BIRTH	SUB_ID	TITLE
1001	Samanaj Bhuiyan Sudul	30-OCT-01	Phy-909	Physics
1005	Nazmul Islam Sizan	23-DEC-01	Phy-909	Physics
1004	Israt Jahan Rumpa	10-AUG-02	Phy-909	Physics

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'

NAME	CLASS_NAME	DATE_OF_BIRTH	SUB_ID	TITLE
Masud Rana	Seven	14-JAN-04	Rln-708	Religion
Masud Rana	Seven	14-JAN-04	Ban-701	Bangla 1st paper

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)

ID	NAME	SUB_ID	TITLE	CLASS_NAME	YEAR
101	Md. Samiul Islam	Math-505	Math	Five	2014
102	Mrs. Islam	Ban-401	Bangla	Four	2014
102	Mrs. Islam	Ban-701	Bangla 1st paper	Seven	2011
405	Md. Sahid Islam	Eng-604	English 2nd paper	Six	2016
408	Al-Amin Hossian	Sco-405	Social Science	Four	2016
409	Md. Saifur Rahman	Phy-909	Physics	Ten	2010
501	Md. Nazmul Islam	Bio-911	Biology	Ten	2014
501	Md. Nazmul Islam	Sci-805	Science	Eight	2015
503	Md. Pervez Hossian	Eng-803	English 1st paper	Eight	2013
602	Md. Aminul Islam	Che-908	Chemistry	Nine	2011
602	Md. Aminul Islam	Ict-809	Information Technology	Eight	2012
705	Mehedi Hasan	Rln-708	Religion	Seven	2009

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

ID	NAME
101	Md. Samiul Islam

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'

ROOM_NUMBER	CAPACITY	CLASS_NAME	SEAT	ROOM_NUMBER
003	40	Child	40	003

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;

YEAR	INSTR_COUNT
2009	1
2010	1
2011	2
2013	1
2014	2
2015	1
2016	2

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

ID	NAME	DATE_OF_BIRTH
805	Jesmin Akter	29-JUL-03
809	Umme Sydati Subha	15-NOV-03

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

YEAR	INSTR_COUNT
2011	2
2014	2
2016	2

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)

NAME
Md. Samiul Islam

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)

ID	NAME
409	Md. Saifur Rahman
501	Md. Nazmul Islam
503	Md. Pervez Hossian
602	Md. Aminul Islam
705	Mehedi Hasan

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')

COUNT(DISTINCTID)
3

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

NO_OF_ID
3

Avg:

Query:

Finding the average salary of the teachers

SQL:

select avg(salary) as avg_sal

from teacher

AVG_SAL
23666.666666666666666666666666666666667

Max:

Query:

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

select max(capacity)

from classroom

MAX(CAPACITY)
100

Min:

Query:

Finding the minimum number of seat of every class.

SQL:

select class_name,min(seat)

from class

group by class_name

CLASS_NAME	MIN(SEAT)
Child	40
Eight	25
Five	35
Four	35
Nine	25
One	40
Seven	30
Six	30
Ten	25
Three	35
Two	35

Sum:

Query:

Total number of seat in the school.

SQL:

select sum(seat)

from class

SUM(SEAT)
355

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)

SUB_ID
Ban-401
Bio-911
Eng-604
Math-505
Sco-405

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)

SUB_ID
Ban-401
Bio-911
Math-505

5.10 Modification(insert, delete, update):

Insertion:

Query:

Inserting a students information.

SQL:

insert into student

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

```
1 row(s) inserted.

0.01 seconds
```

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;

```
5 row(s) updated.

0.00 seconds
```

EDIT	ID	NAME	SALARY
📝	101	Md. Samiul Islam	30000
📝	102	Mrs. Islam	28000
📝	405	Md. Sahid Islam	25000
📝	408	Al-Amin Hossian	25000
📝	409	Md. Saifur Rahman	25300
📝	501	Md. Nazmul Islam	24200
📝	503	Md. Pervez Hossian	24200
📝	602	Md. Aminul Islam	22000
📝	705	Mehedi Hasan	19800
		row(s) 1 - 9 of 9	

Delete:

Query:

Deleting a row from teacher table.

SQL:

delete from teacher

where salary<20000

```
1 row(s) deleted.

0.03 seconds
```

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 (<u>Room_Number</u>, Capacity) is a super key.

• In Class relation <u>Class_Name</u> is a primary key.

• In Subject relation(<u>Sub_ID</u>, Title) is super key.

• In Teacher relation (<u>ID</u>, Name) is a super key.

• In Section relation (<u>Sub_ID</u>, <u>Sec_ID</u>, <u>Year</u>) is a primary key.

• In Teaches relation (<u>ID</u>, <u>Sub_ID</u>, <u>Sec_ID</u>, <u>Year</u>) is a primary key.

• In Student relation <u>ID</u> is a primary key.

• In Routine relation (<u>Routine_ID</u>, <u>Day</u>, <u>Start_Hr</u>) 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 (<u>Sub_ID</u>, <u>Sec_ID</u>, <u>Year</u>, 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 (<u>ID</u>, <u>Sub_ID</u>, <u>Sec_ID</u>, <u>Year</u>)

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

7.7. Student (<u>ID</u>, 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.