**SCHOOL DATABASE MANAGEMENT SYSTEM**

**DBMS FA-4**

**II Year (III semester)**

*of*

**Bachelor of Technology**

in

**Computer Science & Engineering**

By:

Vanshita Kakkar

Maninder Kaur

Radhika Gupta

Dolly Gupta

Submitted to:

Mrs Inderpreet Kaur



**Chitkara University Institute of Engineering and Technology**

**Year (2020 -2024)**

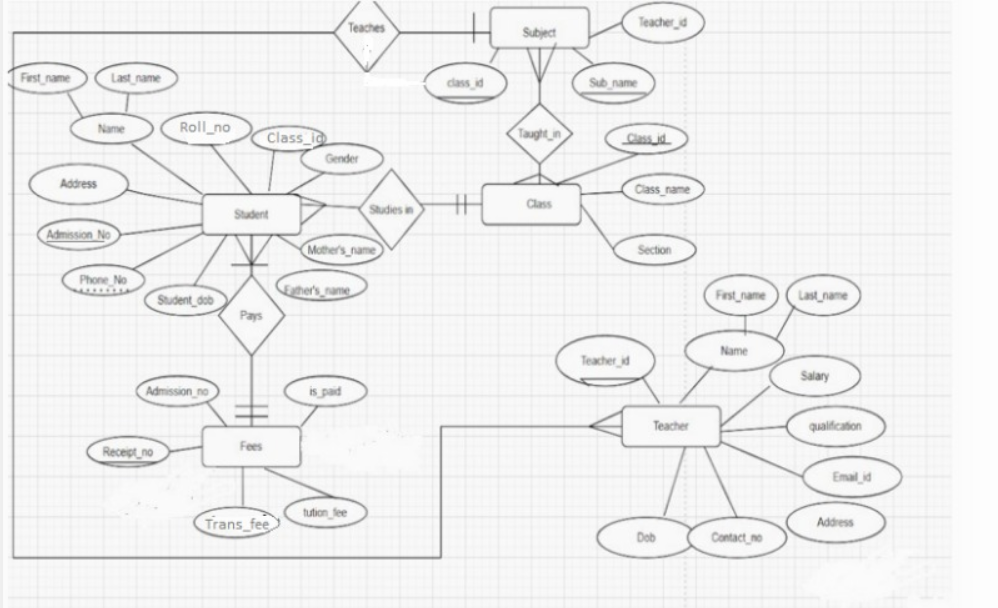
**School Database Management System**

**Description of the project:**.

The project is “SCHOOL MANAGEMENT SYSTEM”. This package once developed will help school to manage various details pertaining to its students. This package is basically developed for the authorities of the school to make their task easier or we can say this package automate their tasks like maintaining students personal details, marinating cash details, printing of receipts. This package helps the administrative and accounts department in maintaining the students personal and fees related details. The basis need of this package was to automate the whole procedure of maintaining of student details, earlier it was all done manually.

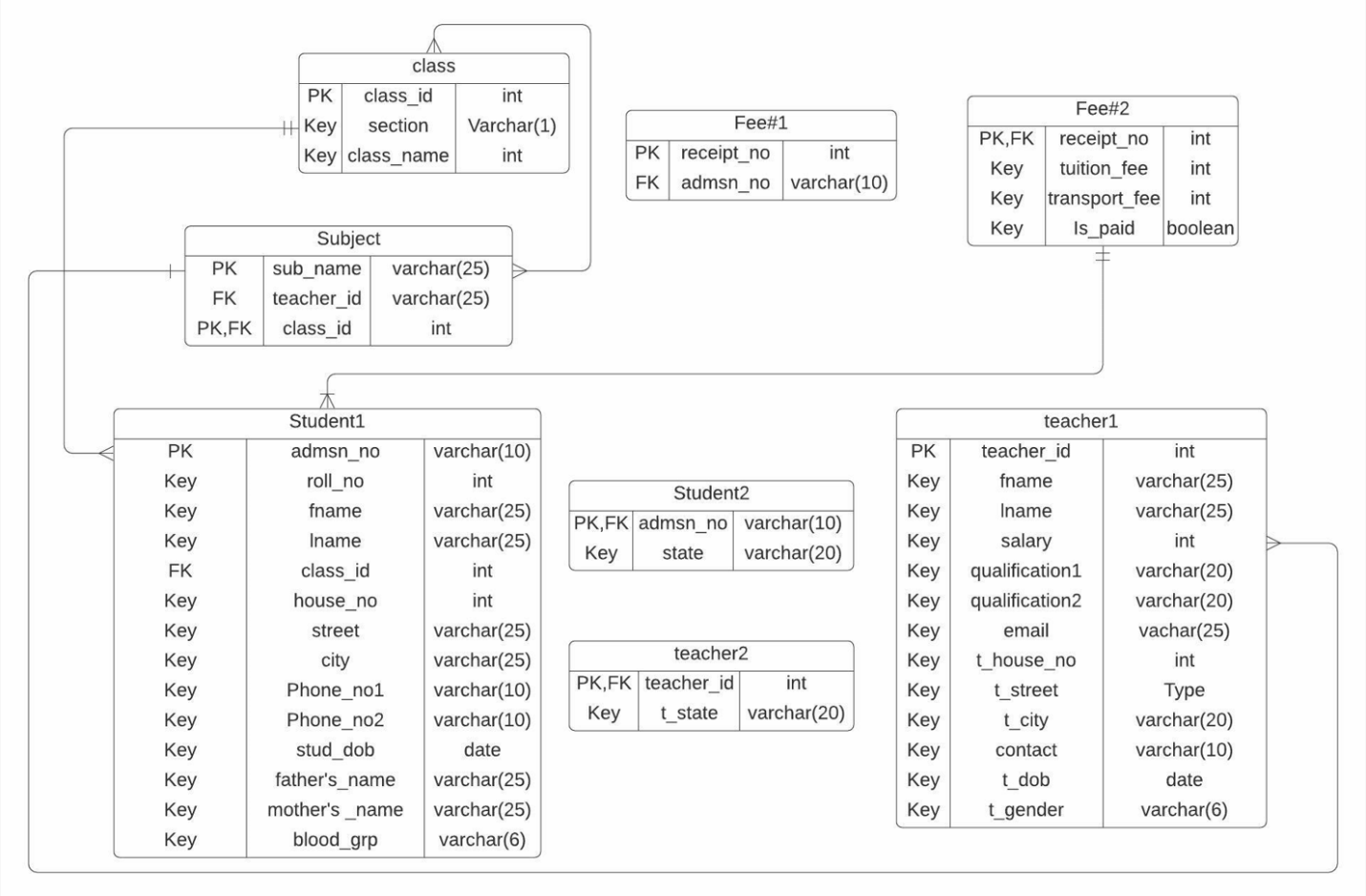
The solution for all problems was to automate the system, automation of the students data maintenance would reduce the manpower, man days will result in accurate data and above all increases the efficiency of the concerned department.

**ER Diagram**

****

**Database**

Schema Representation After Normalisation



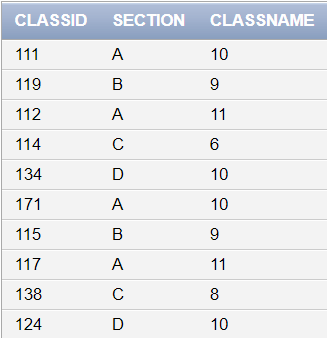
Tables

1)Class

create table class (classid int primary key ,

section varchar(1),

classname int);



2)Student#1

create table student#1 (rollno int,

fname varchar (30)not null,

lname varchar(10) ,

classid int references class(classid),

street varchar(40),

city varchar (10),

adm\_no int primary key,

father\_name varchar(30),

mother\_name varchar(30),

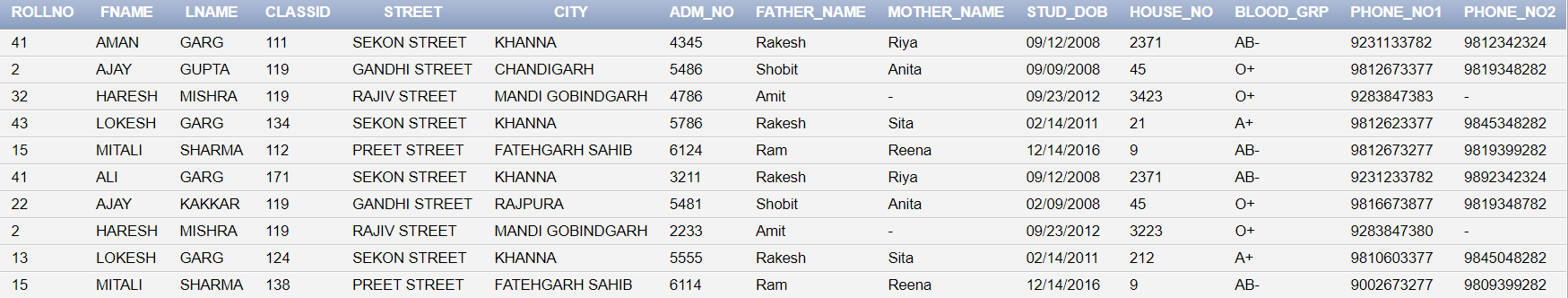
stud\_dob DATE,

house\_no int,

blood\_grp varchar(5),

phone\_no1 number(10),

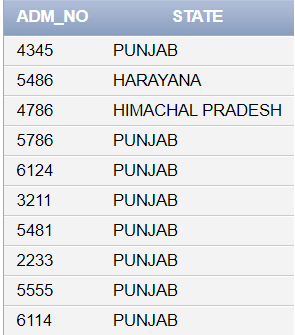
phone\_no2 number(10));



3)Student#2

create table student#2 (adm\_no int references student#1(adm\_no) unique,

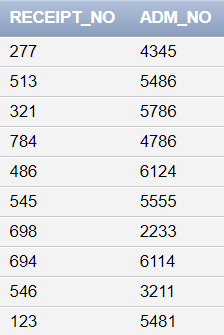
state varchar(20));



4)Fee#1

create table fee#1 (receipt\_no int primary key ,

adm\_no int references student#1(adm\_no) unique);



5)Fee#2

CREATE TABLE Fee#2(reciept\_no int references fee#1(receipt\_no) primary key,

is\_paid char check (is\_paid in('y','n')) ,transport\_fee int,

tuition\_fee int);



6)Teacher#1

create table teacher#1(teacher\_id int primary key,

Fname varchar(30) not null,

Lname varchar(30),

salary number not null,

qualification1 varchar2(10) not null,

qualification2 varchar2(10) not null,

contact\_no number(10) unique,

email\_id varchar2(20) unique,

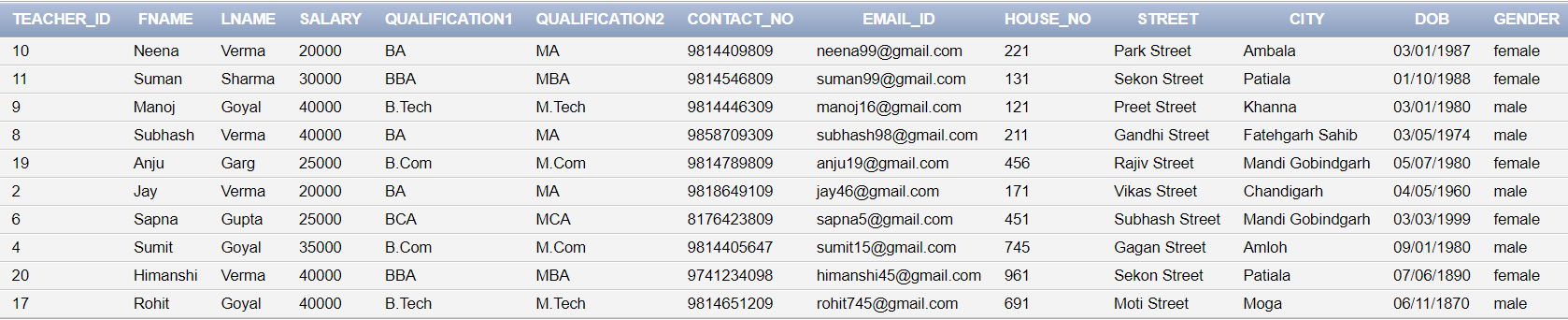
house\_no int not null,

street varchar(30),

city varchar(30),

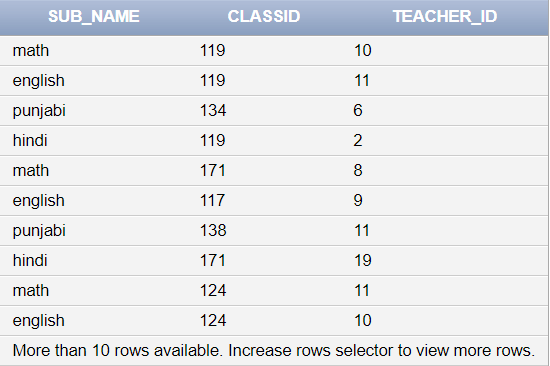
dob date not null,

gender varchar2(10) not null);



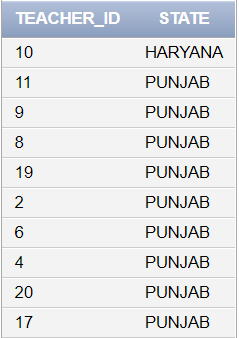
7)Subject

create table subject(sub\_name varchar(10), classid int references class(classid), teacher\_id int references teacher#1(teacher\_id),primary key(sub\_name,classid));



8)Teacher#2

create table teacher#2(teacher\_id int references teacher#1(teacher\_id) primary key, state varchar(20));



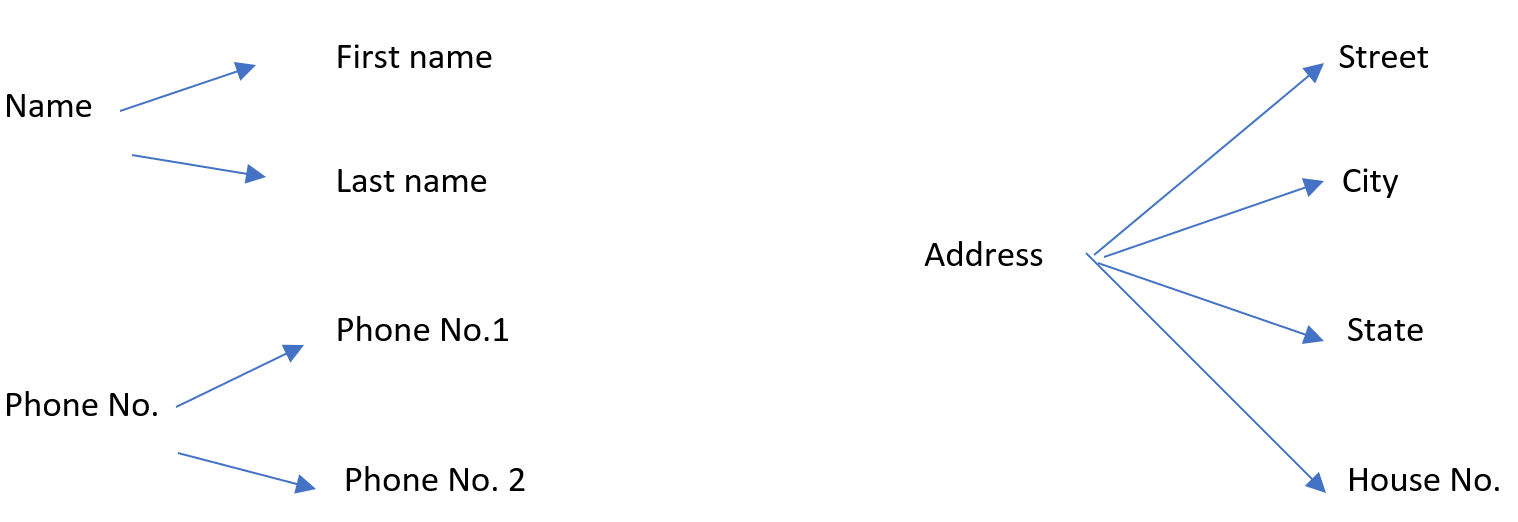
**Normalization**

**Student Table**

Student table before normalisation

|  |
| --- |
| Name |
| Roll no. |
| Class \_ id |
| Phone no. |
| Address |
| Adm\_ no. |
| Stud\_dob |
| Father\_name |
| Mother\_name |
| Blood\_group |

**To convert into 1NF**



Student Table after 1NF

|  |  |
| --- | --- |
| STUDENT | |
| Roll no. | Mother name |
| Class \_ id | Blood group |
| Phone no.1 | Street |
| Phone no. 2 | House \_no. |
| Adm\_ no. | city |
| Stud\_dob | State |
| F\_name | L\_name |
| Father\_name | Mother\_name |

Now Student table is in 1NF as all the attributes has atomic values.

**TO CONVERT INTO 2NF**

**DEPENDENCIES**

Roll\_no Classid ---> Fname , lname , street ,adm\_no. ,phone\_no.1 , phone\_no.2 ,house\_no mother\_name, father\_name , blood\_grp, city.

City ---> state

Adm\_no ------>roll.no class id

Candidate Key: Roll\_No class\_id ,adm\_no

Primary Key:adm\_no

There is no partial Dependency as there is no subset of any candidate key that derives non-prime attribute.

So the student table is in 2NF form.

**TO CONVERT INTO 3NF**

Adm\_no → city→ state

Adm\_no → state

This relation exists in the student table. So student table is not in 3NF.

So student table is decomposed into student#1 and student#2.

**STUDENT#1**

|  |  |
| --- | --- |
| Roll no. | Fname |
| Class \_ id | Lname |
| Phone no.1 | Street |
| Phone no.2 | Father\_name |
| Adm\_ no. **PK** | City |
| Stud\_dob | House no. |
| Blood\_group | Mother\_name |

**Student#2**

|  |
| --- |
| Adm\_no **PK,FK** |
| State |

**DEPENDENCIES AFTER DECOMPOSITION OF STUDENT TABLE**

Student#1

Roll\_no Classid ---> Fname , lname , street ,adm\_no. ,phone\_no.1 , phone\_no.2 ,house\_no mother\_name, father\_name , blood\_grp, city.

Adm\_no ------>roll.no class id

Student#2

Adm\_no-->state

**BCNF :**

As in all functional dependencies, there is candidate key on left hand side.So student#1 and student#2 are in BCNF.

**FEE TABLE :**

**Fee**

|  |
| --- |
| Receipt\_no **PK** |
| Adm\_no **FK** |
| tuition\_fee |
| transport\_fee |
| is\_paid |

**TO CONVERT INTO 1NF**

Fee table is in 1NF because all the attributes of fee have atomic values so there’s no need to further bifurcate the attributes, multivalued attributes are not in picture.

**TO CONVERT INTO 2NF**

**Dependencies :**

receipt\_no adm\_no

adm\_no is\_paid, transport\_fee, tuition\_fee

Candidate key = receipt\_no = Primary Key

As the candidate key is singleton candidate key, so imagining a partial dependency is meaningless that’s why the table is in 2NF.

**TO CONVERT INTO 3NF**

receipt\_no adm\_no is\_paid, transport\_fee, exam\_fee, tuition\_fee

receipt\_no is\_paid, transport\_fee, exam\_fee, tuition\_fee

The existence of the above written dependency makes the table to not be in 3NF that’s why the following bifurcation is provided :

|  |
| --- |
| **Fee#1** |
| receipt\_no **PK** |
| adm\_no **FK** |

|  |
| --- |
| **Fee#2** |
| receipt\_no **PK,FK** |
| is\_paid |
| transport\_fee |
| tution\_fee |

The transitivity has been eliminated and thus the tables are in 3NF.

**DEPENDENCIES**

**Fee#1 :**

reciept\_no adm\_no

**Fee#2 :**

receipt\_no is\_paid, transport\_fee, exam\_fee, tuition\_fee

**BCNF :**

As in all functional dependencies, there is candidate key on left hand side.So Fee#1 and Fee#2 are in BCNF.

**Teacher Table :**

**Teacher**

|  |
| --- |
| teacher\_id **PK** |
| teacher\_name |
| salary |
| qualification |
| email\_id |
| address |
| contact\_no |
| t\_dob |
| gender |

**TO CONVERT INTO 1NF**

f\_name

teacher\_name

l\_name

house\_no

Address street

city

state

qualification1

Qualification

qualification2

|  |
| --- |
| **Teacher** |
| teacher\_id **PK** |
| f\_name |
| l\_name |
| salary |
| qualification1 |
| qualification2 |
| contact\_no |
| street |
| email\_id |
| house no |
| city |
| state |
| dob |
| gender |

1NF has been achieved.

**Dependencies :**

teacher\_id f\_name, l\_name, salary, contact\_no, gender, dob, qualification1, qualification2 ,email\_id, city

email\_id contact\_no, teacher\_id

contact\_no email\_id

city state

**2NF :**

**DEPENDENCIES**

Candidate Key :Contact\_no, email\_id, teacher\_id

Primary Key : teacher\_id

As the candidate key is singleton candidate key so there is no meaning for partial dependencies. Teacher table is in 2NF.

**3NF :**

teacher\_idcitystate

teacher\_idstate

The above mentioned relations are transitive in nature thus it leads to the violation of 3NF form. The following bifurcation will end this transitivity and thus leading to the proper 3NF form.

|  |
| --- |
| **Teacher#1** |
| teacher\_id **PK** |
| f\_name |
| l\_name |
| salary |
| qualification1 |
| qualification2 |
| contact\_no |
| email\_id |
| city |
| dob |
| house\_no |
| street |
| gender |

|  |
| --- |
| **Teacher#2** |
| teacher\_id **PK** |
| state |

3NF form has been achieved.

**DEPENDENCIES AFTER 3NF:**

**Teacher#1**

teacher\_id f\_name, l\_name, salary, contact\_no, gender, dob, qualification1, qualification2 ,email\_id, city

email\_id contact\_no, teacher\_id

contact\_no email\_id

**Teacher#2**

teacher\_id state

**BCNF :**As in all functional dependencies, there is candidate key on left hand side.So Fee#1 and Fee#2 are in BCNF.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SUBJECT TABLE**

|  |
| --- |
| **Subject** |
| Sub\_name  **PK** |
| Class\_id **PK,FK** |
| teacher\_id **FK** |

**1NF :** Subject table is already in 1NF as there is absency of multivalued attributes and atomic states of all the attributes are present in the table.

**Dependencies :**

Sub\_name class\_id teacher\_id

Candidate Key = Sub\_name, Class\_id = Primary Key

**2NF :** There is no partial Dependency as there is no subset of any candidate key that derives non-prime attribute.

**3NF :** There is no transitivity as well so the table is already in 3NF.

**BCNF :**As in all functional dependencies, there is candidate key on left hand side.So Subject table is in BCNF.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Table**

|  |
| --- |
| **Class** |
| class\_id **PK** |
| section |
| class\_name |

**Dependencies :**

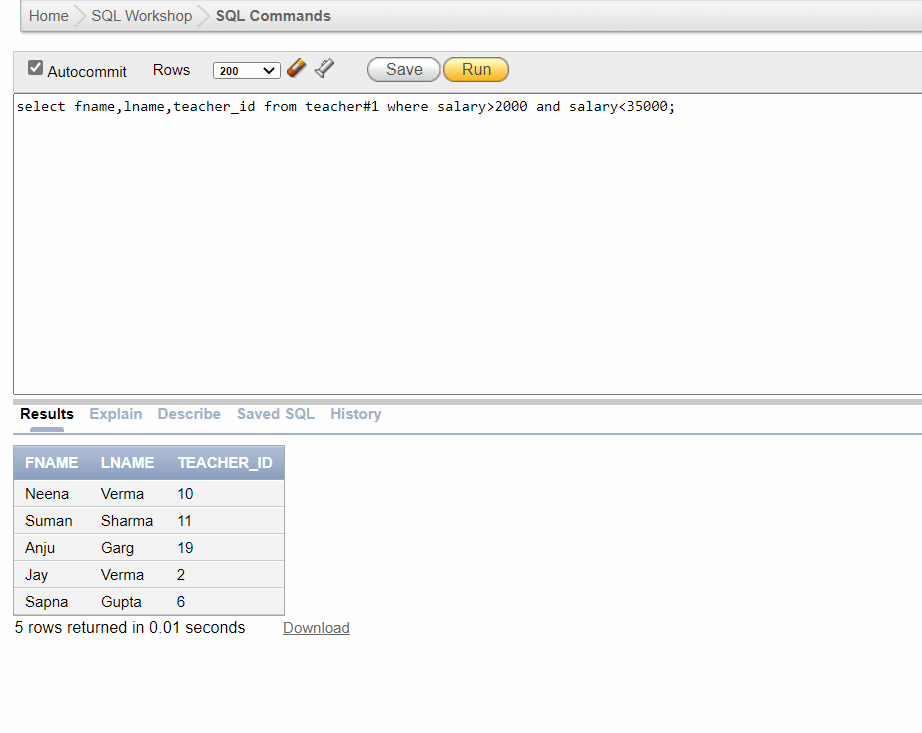
Class\_id section, class\_name

Candidate Key = Class\_id = Primary Key

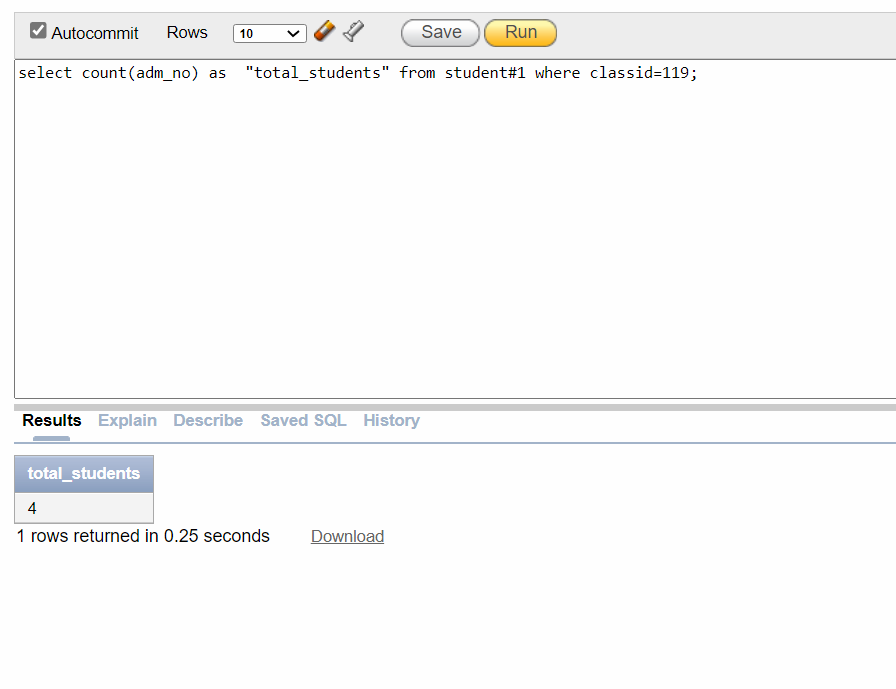
The fact that the table is in BCNF eliminates the need to check for lower normal forms.

**Queries**

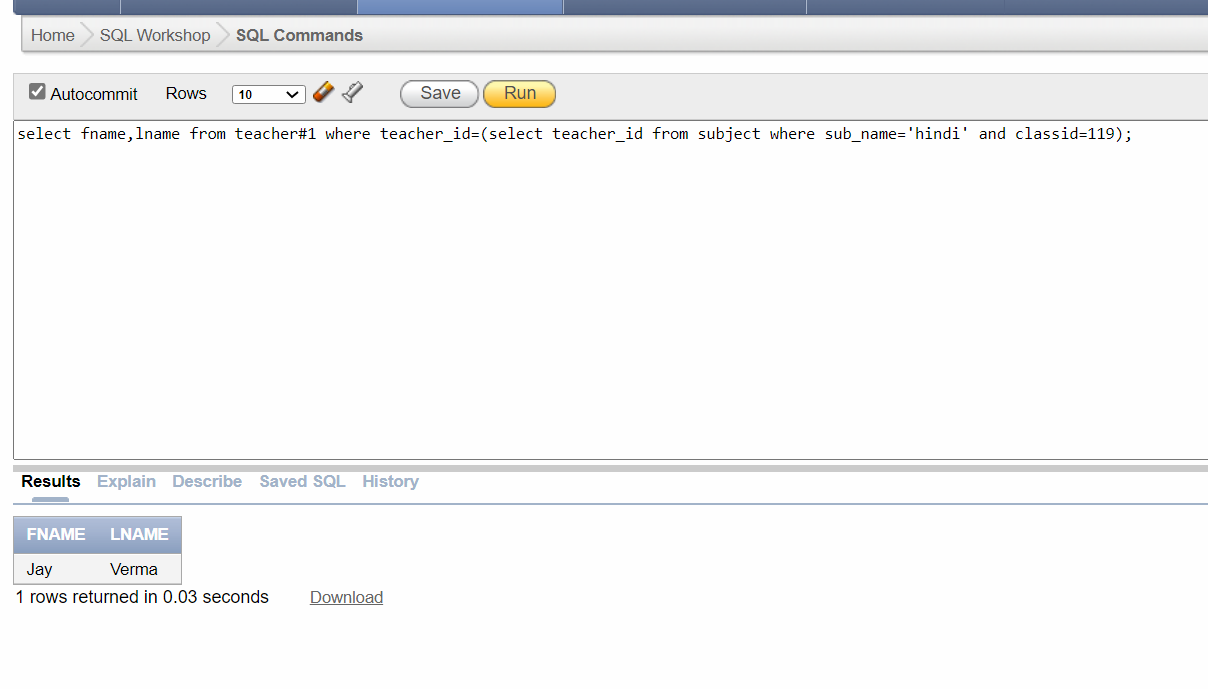
1)Find fname ,lname ,teacher\_id of teacher whose salary is greater than 20,000 but less than 35000



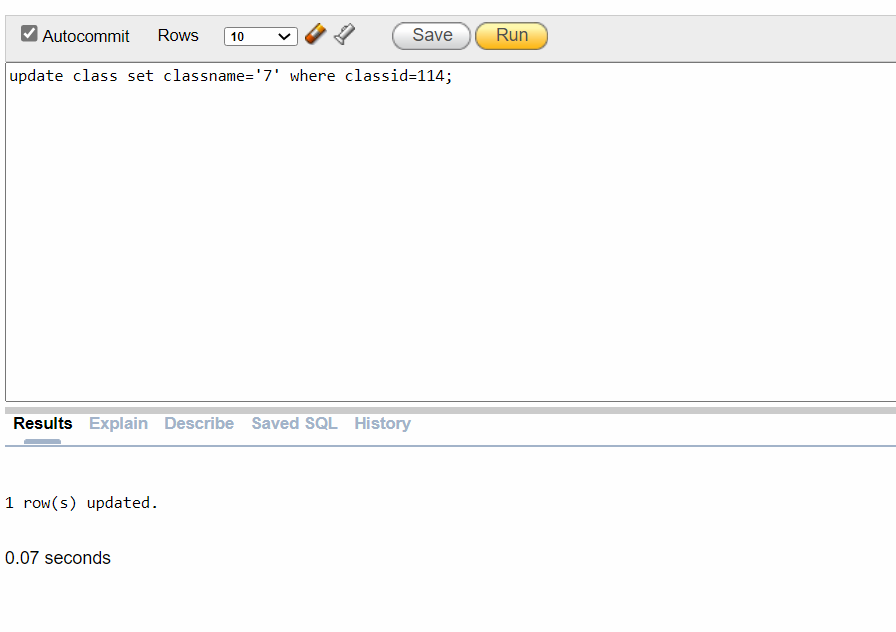
2)Find out number of students in class which have classid=119



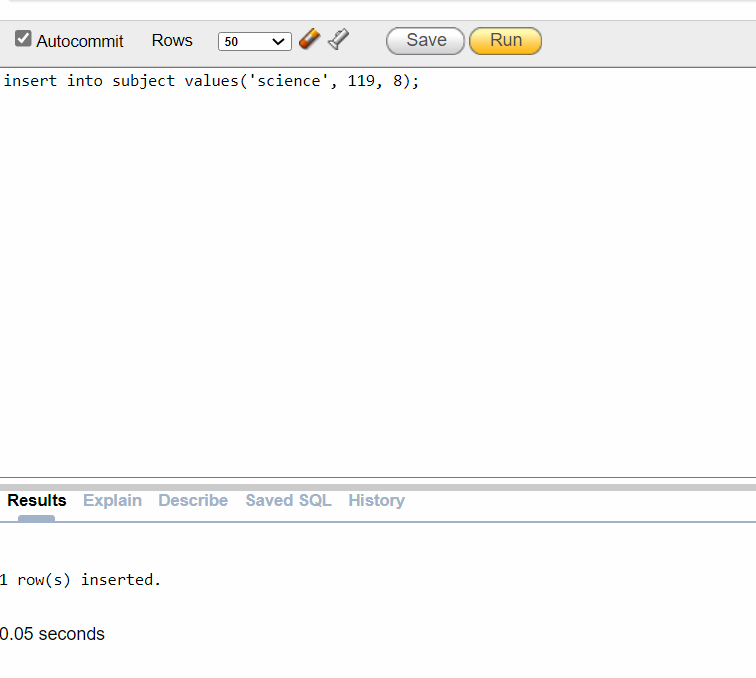
3)Find out the teacher name who teaches hindi to class having classid=119.

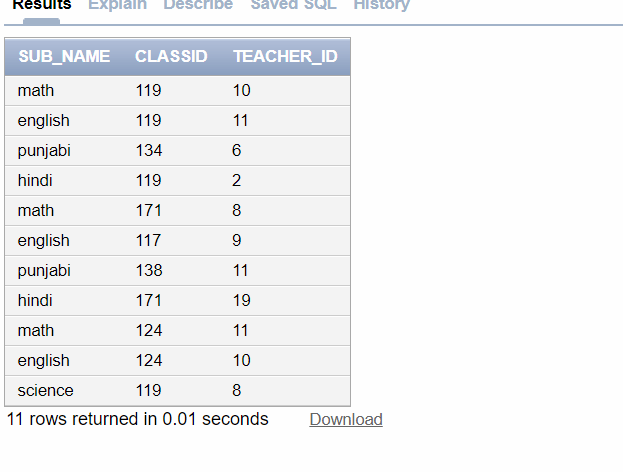


4)Update the class name which has classid 114.

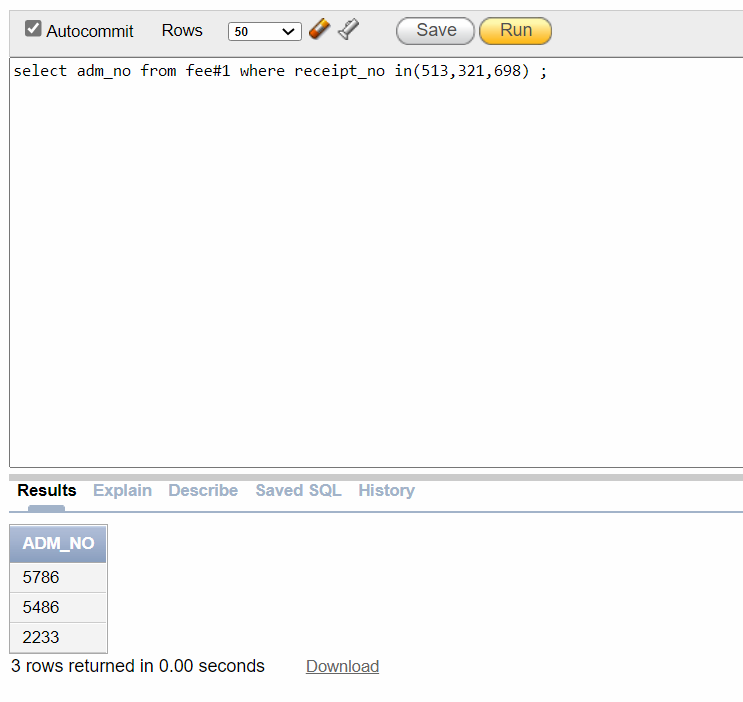


5)Add science subject in class having class\_id 119 and teacher with teacher\_id 8 teaches.

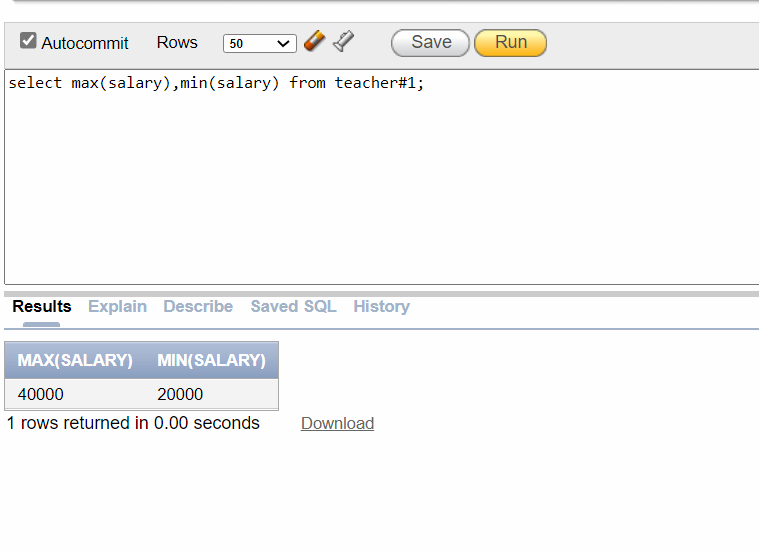




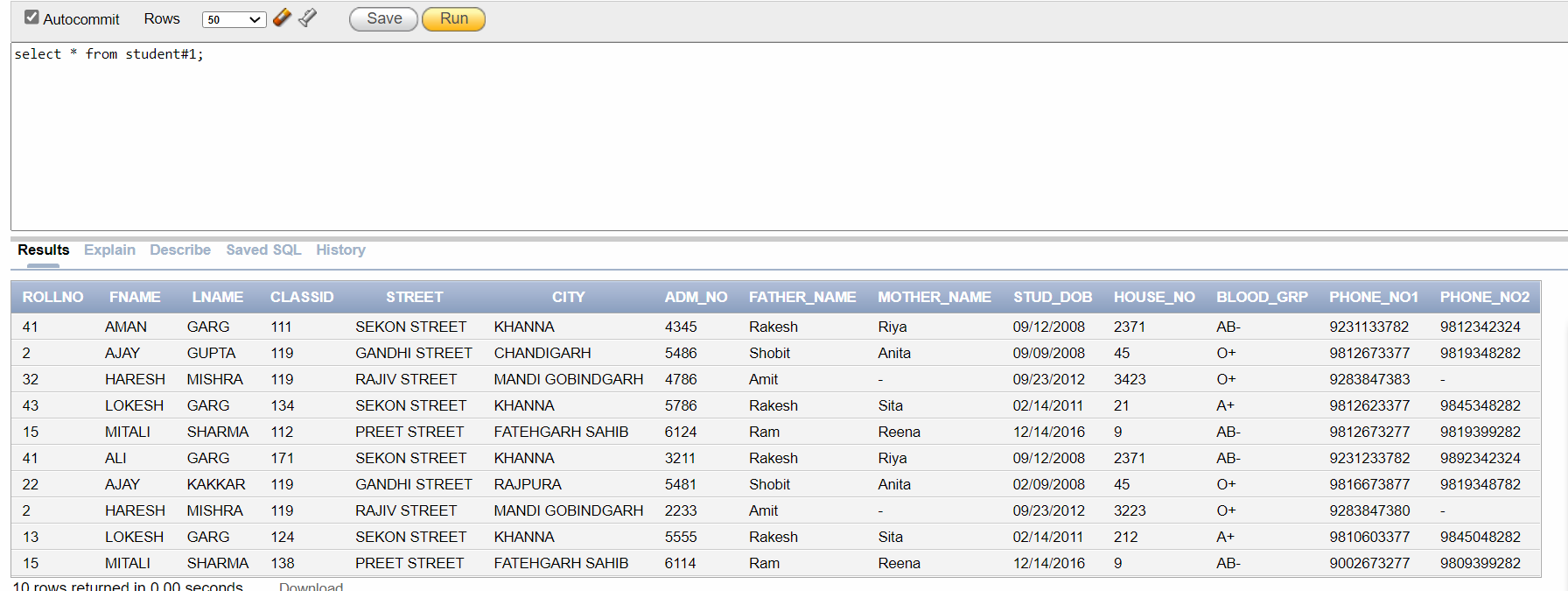
6) Find the students’ admission no who have receipt no 513,321,698.



7) Find max and min salary of teacher.



8)Find all details of all students.



**Work division table**

|  |  |
| --- | --- |
| **Student Name** | **Work done** |
| **Vanshita Kakkar** | **Normalization, Queries** |
| **Maninder Kaur** | **Schema Representation,Queries** |
| **Radhika Gupta** | **ER Diagram,Queries** |
| **Dolly Gupta** | **ER Diagram, Tables** |