**SCHOOL**

**MANAGEMENT**

**SYSTEM**

**VIKAS PANDEY (2019BCS-071)**

**SHUBHAM GANGWAR (2019BCS-060)**

**GAURAV SHRIWASTAV (2019BCS-076)**

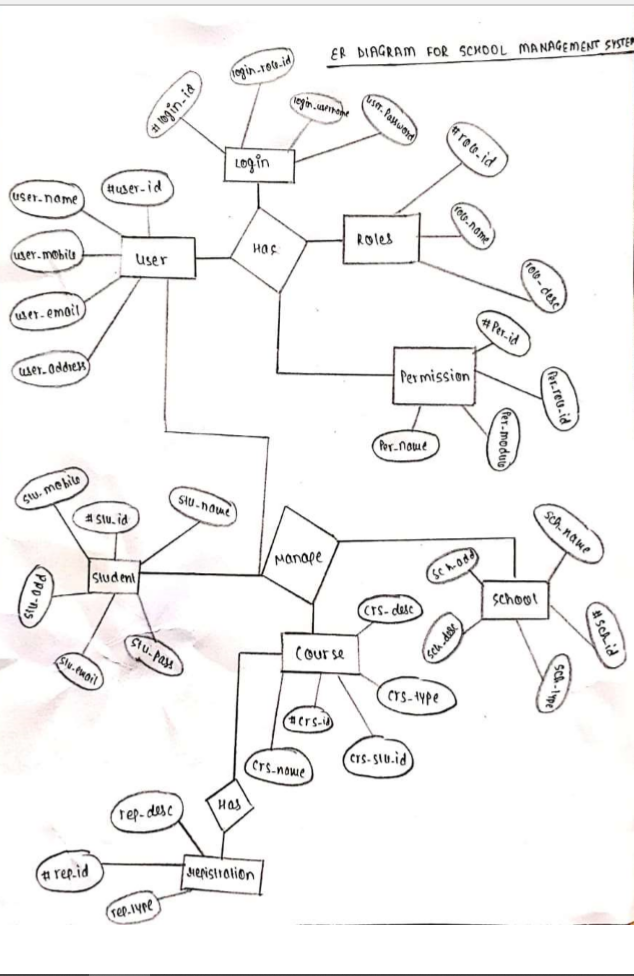
**. To create a database management system to curb and minimize the day to day problems that occur in the manual management systems of the schools. The project is carried out to minimize the errors that may occur in the process of storing and tallying the data at hand. The database contains the following data:**

**. Description of Schools Management System Database:**

1. **The details of schools are store into the school’s tables respective with all tables.**
2. **Each entity (Registrations, classes, courses, students, schools) contains primary key and unique keys.**
3. **The entity classes, courses have binded with schools, students’ entities with foreign key.**
4. **There is one-to-one and one-to-many relationships available between courses, teachers, registrations, schools.**
5. **All the entities schools, courses, classes, registrations are normalized and reduce duplicacy of records.**
6. **We have implemented indexing on each table for school management system tables for fast query execution.**

**. Schools Management System entities and their attributes:**

1. **Schools Entity:** Attributes of schools are school id, school name, school type, school description.
2. **Students Entity:** Attributes of students are student id, student college id, student name, student mobile, student email, student username, student password, student address.
3. **Classes Entity:** Attributes of classes are class id, class student id, class name, class room, class type, class description.
4. **Teacher Entity:** Attributes of teachers are teacher id, teacher college is, teacher name, teacher mobile, teacher email, teacher username, teacher password, teacher address.
5. **Courses Entity:** Attributes of courses are course id, course student id, course registration, course name, course type, course year, course description.
6. **Registration Entity:** Attributes of registrations are registration id, registration student id, registration course id, registration name, registration type, registration number, registration date, registration description.

****

* **Functional Dependency**
* A functional dependency (FD) is a relationship between two attributes, typically between the PK and other non-key attributes within a table**.**
* is functionally dependent on attribute X (usually the PK), if for every valid instance of X, that value of X uniquely determines the value of Y.
* **User**

User id+ user name+ user mobile no+ user email+ user address

* **Login**

Login id+ login role id+ login username+ user password

* **Roles**

Role id+ role name

* **Student**

Student name+ student id+ student mobile+ student address+ student email+ student password

* **Courses**

Course name+ Course id+ Course student id+ Course type+ Course desc

* **Registration**

Registration id+ Registration type + Registration desc

* **School**

School id+ school name+ school address

* **Fees**  
  ***student\_id + transaction id*** 🡪 student id + transaction id + fee month  
  ***transaction id*** 🡪 transaction id + fee amount + fee status
* **Classrooms**  
  ***Room\_id***🡪 room id + capacity   
  ***capacity***🡪 capacity + No. of seats
* **TEACHER**

Teacher id+ Name+ subject

**Tables in 1st Normalized Form**

* **Definition**

First normal form (1NF) is a property of a [relation](https://en.wikipedia.org/wiki/Relation_(database)) in a [relational database](https://en.wikipedia.org/wiki/Relational_database). A relation is in first normal form if and only if the [domain](https://en.wikipedia.org/wiki/Data_domain) of each [attribute](https://en.wikipedia.org/wiki/Column_(database)) contains only [atomic](https://en.wikipedia.org/wiki/First_normal_form) (indivisible) values, and the value of each attribute contains only a single value from that domain.

**Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Login\_id** | **Login\_rol\_id** | **Login\_username** | **Login\_pass** |
| **A001** | **2019B01** | **Ajay\_001** | **Xyz** |
| **A002** | **2019B07** | **Dev\_007** | **Abc** |
| **A003** | **2019B09** | **Vijay\_009** | **Def** |
| **A004** | **2019B010** | **Rahul\_010** | **tyew** |
| **A005** | **2019B011** | **Raj\_011** | **yret** |

**Role**

|  |  |
| --- | --- |
| **Role\_id** | **Role\_name** |
| **001** | **Ajay** |
| **007** | **Dev** |
| **009** | **Vijay** |
| **011** | **Raj** |
| **010** | **Rahul** |

**Tables in 2nd Normalized Form**

**Definition:**

* Second normal form (2NF) is a [normal form](https://en.wikipedia.org/wiki/Database_normalization) used in [database normalization](https://en.wikipedia.org/wiki/Database_normalization).
* To qualify for second normal form a relation must:

1. be in [first normal form](https://en.wikipedia.org/wiki/First_normal_form) (1NF)

(2) not have any [non-prime attribute](https://en.wikipedia.org/wiki/Non-prime_attribute) that is [dependent](https://en.wikipedia.org/wiki/Functional_dependency) on any [proper subset](https://en.wikipedia.org/wiki/Proper_subset) of any [candidate key](https://en.wikipedia.org/wiki/Candidate_key) of the relation. **A non-prime attribute of a relation** is an attribute that is not a part of any candidate key of the relation.

**Classroom**

|  |  |  |
| --- | --- | --- |
| Room\_id | Capacity | No.of seat |
| LT-01-001 | 100 | 80 |
| LT-01-002 | 80 | 60 |
| LT-02-008 | 95 | 83 |
| LT-02-006 | 150 | 120 |
| LT-01-004 | 100 | 81 |
| LT-02-010 | 200 | 170 |
| LT-01-005 | 110 | 90 |

**Fees**

|  |  |  |
| --- | --- | --- |
| **Fee Month** | **St\_id** | **Transaction id** |
| January | A001 | BKID001 |
| February | A002 | BKID002 |
| March | A003 | BKID003 |
| April | B001 | BKID004 |
| May | B002 | BKID005 |
| June | B003 | BKID006 |

|  |  |  |
| --- | --- | --- |
| **Fess amount** | **Fees Status** | **Transaction id** |
| 40000 | Paid | BKID005 |
| 40000 | Paid | BKID003 |
| 40000 | Paid | BKID004 |
| 40000 | Paid | BKID001 |

**Teacher**

|  |  |  |
| --- | --- | --- |
| **Teacher id** | **Name** | **Subject** |
| IIIT\_010 | Joydip | Maths |
| IIIT\_020 | Pankaj | Physics |
| IIIT\_030 | Anurag | Electronics |
| IIIT\_040 | Mahesh | Computer |
| IIIT\_011 | Vijay | Data Struct |

**Tables in 3rd Normalized Form**

**Definition**

* Third normal form (3NF) is a [normal form](https://en.wikipedia.org/wiki/Database_normalization) that is used in [normalizing](https://en.wikipedia.org/wiki/Database_normalisation) a [database](https://en.wikipedia.org/wiki/Database) design to reduce the duplication of data and ensure [referential integrity](https://en.wikipedia.org/wiki/Referential_integrity) by ensuring that:
* (1) the entity is in [second normal form](https://en.wikipedia.org/wiki/Second_normal_form)
* (2) no non-prime (non-key) attribute is transitively dependent of any key i.e. no non-prime attribute depends on other non-prime attributes. All the non-prime attributes must depend on the primary key only.

**Student**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Id** | **DOB** | **Mobile** | **City** | **State** | **Email** |
| Ajay | 01 | 04/10/01 | 8765542 | Thane | MH | [abc@gmail.com](mailto:abc@gmail.com) |
| Dev | 07 | 20/4/00 | 9642434 | Bandra | MH | [yr@gmail.com](mailto:yr@gmail.com) |
| Vijay | 09 | 10/3/00 | 9542445 | Mumbai | MH | [xz@gmail.com](mailto:xz@gmail.com) |
| Rahul | 10 | 21/7/01 | 8545414 | Andheri | MH | [a@gmail.com](mailto:a@gmail.com) |
| Raj | 11 | 24/08/00 | 9643453 | Worli | MH | [xs@gmail.com](mailto:xs@gmail.com) |
| Roy | 15 | 31/12/01 | 8544422 | Churchgate | MH | [b@gmail.com](mailto:b@gmail.com) |

**Registration**

|  |  |
| --- | --- |
| **Reg\_id** | **Reg\_type** |
| A001 | Abc |
| B001 | Xyz |
| C001 | Zax |
| A002 | Pox |
| A004 | Rey |

**Courses**

|  |  |  |
| --- | --- | --- |
| **Course\_id** | **Course\_name** | **Crs\_stu\_id** |
| C001 | BCS | S001 |
| C002 | IT | S002 |
| C003 | MBA | S003 |
| C004 | ECE | S004 |
| C005 | MECH | S005 |

**To ensure lossless decomposition, the following 3**

**conditions must hold:-**

* Join of attributes of decomposed table must consist of all the attributes of original table.
* Intersection of decomposed tables must not be null.
* The common attribute amongst the decomposed tables must be the primary key of either table or both.

**THANKS**