**NATIONAL INSTITUE OF TECHNOLOGY, HAMIRPUR**

**CS-312: DATABASE MANAGEMENT SYSTEM**

**TERM PROJECT**

**COLLEGE DATA MANAGEMENT**

**Submitted to: Prof. Vijay Kumar Chahar**

**Made By:**

**CSE (Dual), Fifth Semester**

**GROUP 12:**

* **Shivam Pathak 195509**
* **Vani Sharma 195510**
* **Anmol Madaik 195513**
* **Deepanshu Sharma 195520**
* **Aman Verma 195522**

**INDEX**

1. INTRODUCTION 3

2. ER DIAGRAM 4

3. RELATIONAL SCHEMA 11

4. SQL QUERIES 12

**INTRODUCTION**

The college data management contains data related to the college, i.e. various departments in the college, courses in a department, the instructors teaching that course and the students studying various courses. The E-R Diagram and the Relational Schema for the college data has also been made.

**ER DIAGRAM**

The E-R data model employs three basic concepts: entity sets, relationship sets, and attributes

1. ENTITY SETS

An entity is a “thing” or “object” in the real world that is distinguishable from all other objects. An entity has a set of properties, and the values for some set of properties may uniquely identify an entity. An entity set is a set of entities of the same type that share the same properties, or attributes. Each entity has a value for each of its attributes. A database thus includes a collection of entity sets, each of which contains any number of entities of the same type.

2. RELATIONSHIP SETS

A relationship is an association among several entities. A relationship set is a set of relationships of the same type. The association between entity sets is referred to as participation. A relationship instance in an E-R schema represents an association between the named entities in the real-world enterprise that is being modeled. The function that an entity plays in a relationship is called that entity’s role. A relationship may also have attributes called descriptive attributes. The number of entity sets that participate in a relationship set is the degree of the relationship set. A binary relationship set is of degree 2; a ternary relationship set is of degree 3.

3. ATTRIBUTES

An attribute of an entity set is a function that maps from the entity set into a domain. For each attribute, there is a set of permitted values, called the domain, or value set, of that attribute. The attribute values describing an entity constitute a significant portion of the data stored in the database. An attribute, as used in the E-R model, can be characterized by the following attribute types:

* Simple and composite attributes. Simple attributes cannot be divided into subparts. Composite attributes can be divided into subparts (that is, other attributes).
* Single-valued and multivalued attributes. Singled value attributes are attributes having a single value. Attributes that have more than one value are called multivalued attributes. Where appropriate, upper and lower bounds may be placed on the number of values in a multivalued attribute.
* Derived attribute. The value for this type of attribute can be derived from the values of other related attributes or entities. The value of a derived attribute is not stored but is computed when required.

E-R diagram can express the overall logical structure of a database graphically. E-R diagrams are simple and clear—qualities that may well account in large part for the widespread use of the E-R model.

An E-R diagram consists of the following major components:

* Rectangles divided into two parts represent entity sets.
* Diamonds represent relationship sets.
* Undivided rectangles represent the attributes of a relationship set. Attributes that are part of the PRIMARY KEY are underlined.
* Lines link entity sets to relationship sets.

**N**

**M**

**COURSES**

**ATTENDS**

**STUDENT**

**M**

**TEACHES**

**HAS**

**M**

**M**

**M**

**TEACHES**

**1**

**1**

**IN**

**1**

**N**

**INSTRUCTOR**

**DEPARTMENT**

**BELONGS**

**M**

**1**

Entities and their attributes used in this E-R diagram are:

* **STUDENT**
  + ROLL NUMBER
  + NAME
  + DATE OF BIRTH
  + MOBILE NUMBER
  + ADDRESS
* **INSTRUCTOR**
  + ID
  + NAME
  + DATE OF BIRTH
  + MOBILE NUMBER
  + ADDRESS
* **DEPARTMENT**
  + NAME
  + HOD
  + NUMBER OF STUDENTS
  + NUMBER OF COURSES
* **COURSES**
  + COURSE ID
  + COURSE NAME

Relationships used in this E-R diagram are:

* ATTENDS (Many to Many from STUDENT to COURSES)
* TEACHES (Many to Many from STUDENT to INSTRUCTOR)
* BELONGS (Many to One from INSTRUCTOR to DEPARTMENT)
* IN (Many to Many from COURSES to DEPARTMENT)
* TEACHES (Many to One from COURSES to INSTRUCTOR)
* HAS (Many to One from STUDENT to DEPARTMENT)

**RELATIONAL SCHEMA**

A relational schema is a set of relational tables and associated items that are related to one another. It is the basic information describing a table and its relation. It is the logical definition of a table. Relation schema defines what the name of the table is. This includes the attributes of the table, the PRIMARY KEY, the referenced Foreign Keys of the table and the relations of the table.

Tables used in Relational Schema:

* **STUDENT**(ROLLNO, NAME, DATE\_OF\_BIRTH, MOBILENO, ADDRESS, *DEPARTMENT)*
* **COURSES**(ID, NAME, *INSTRUCTOR*, *DEPARTMENT)*
* **ATTENDS**(*ROLLNO, COURSE\_ID*)
* **INSTRUCTOR**(ID, NAME, DATE\_OF\_BIRTH, MOBILENO, ADDRESS, *DEPARTMENT)*
* **TEACHES**(*ROLLNO, INSTRUCTOR\_ID*)
* **DEPARTMENT**(NAME, HOD, NUMBER\_OF\_STUDENTS, NUMBER\_OF\_COURSES)

**STUDENT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ROLLNO | NAME | DATE\_OF\_BIRTH | MOBILENO | ADDRESS | DEPARTMENT |

**COURSES**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAME | INSTRUCTOR | DEPARTMENT |

**ATTENDS**

|  |  |
| --- | --- |
| ROLLNO | COURSE\_ID |

**INSTRUCTOR**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | NAME | DATE\_OF\_BIRTH | ADDRESS | MOBILENO | DEPARTMENT |

**TEACHES**

|  |  |
| --- | --- |
| ROLLNO | INSTRUCTOR\_ID |

**DEPARTMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | HOD | NUMBER\_OF\_STUDENTS | NUMBER\_OF\_COURSES |

**SQL QUERIES**

IBM developed the original version of SQL, originally called Sequel, as part of the System R project in the early 1970s. The Sequel language has evolved since then, and its name has changed to SQL (Structured Query Language). Many products now support the SQL language. SQL has clearly established itself as the standard relational database language. In 1986, the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO) published an SQL standard, called SQL-86. ANSI published an extended standard for SQL, SQL-89, in 1989.

The SQL language has several parts:

* Data-definition language (DDL). The SQL DDL provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
* Data-manipulation language (DML). The SQL DML provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database.
* Integrity. The SQL DDL includes commands for specifying Integrity constraints that the data stored in the database must satisfy. Updates that violate Integrity constraints are disallowed.
* View definition. The SQL DDL includes commands for defining views.
* Transaction control. SQL includes commands for specifying the beginning and ending of transactions.
  + Embedded SQL and dynamic SQL. Embedded and dynamic SQL define how SQL statements can be embedded within general-purpose programming languages, such as C, C++, and Java.
* Authorization. The SQL DDL includes commands for specifying access rights to relations and views.

SQL COMMANDS USED TO CREATE REQUIRED DATABASE:

Note: A database dbms is used to store the Database

1. Creating Database:

A database is created to store all the required data using the following SQL statements

* CREATE DATABASE dbms -> Creates a new database dbms
* USE dbms -> Accesses the database dbms

2. Creating Table:

Tables are created with the required attributes according to the relational schema described above:

* CREATE TABLE STUDENT (Roll\_no INT PRIMARY KEY, Name VARCHAR(20), Dob DATE, Mobile\_no BIGINT, Address VARCHAR(40), Department VARCHAR(40));
* CREATE TABLE COURSES(id VARCHAR(10) PRIMARY KEY, Name VARCHAR(20), Instructor VARCHAR(10),Department VARCHAR(40));
* CREATE TABLE ATTENDS (Roll\_no INT, Course\_id VARCHAR(10));
* CREATE TABLE INSTRUCTOR (id VARCHAR(10) PRIMARY KEY, Name VARCHAR(20), Dob DATE, Address VARCHAR(40), Mobile\_no BIGINT, Department VARCHAR(40));
* CREATE TABLE TEACHES (Roll\_no INT, instructor\_id VARCHAR(10));
* CREATE TABLE DEPARTMENT (Name VARCHAR(40) PRIMARY KEY, HOD VARCHAR(20), no\_of\_student INT, no\_of\_courses INT);

3. Adding Foreign Keys:

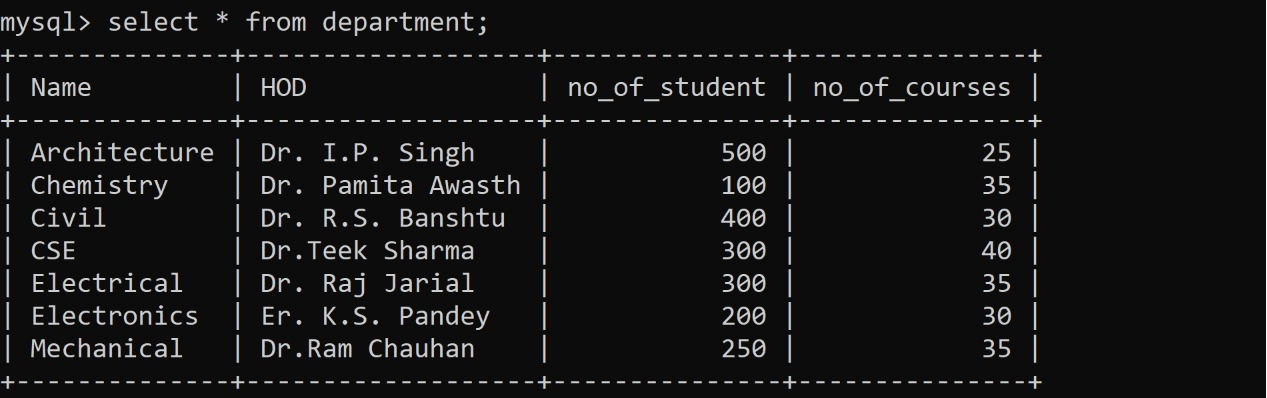
A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables because it REFERENCES the primary key of another table, thereby establishing a link between them.

* ALTER TABLE ATTENDS ADD FOREIGN KEY(Roll\_no) REFERENCES STUDENT(Roll\_no);
* ALTER TABLE ATTENDS ADD FOREIGN KEY(Course\_id) REFERENCES COURSES(id);
* ALTER TABLE COURSES ADD FOREIGN KEY(Instructor) REFERENCES INSTRUCTOR(id);
* ALTER TABLE COURSES ADD FOREIGN KEY(Department) REFERENCES DEPARTMENT(Name);
* ALTER TABLE STUDENTS ADD FOREIGN KEY(Department) REFERENCES DEPARTMENT(Name);
* ALTER TABLE INSTRUCTOR ADD FOREIGN KEY(Department) REFERENCES DEPARTMENT(Name);
* ALTER TABLE TEACHES ADD FOREIGN KEY(Roll\_no) REFERENCES STUDENT(Roll\_no);
* ALTER TABLE TEACHES ADD FOREIGN KEY(instructor\_id) REFERENCES INSTRUCTOR(id);

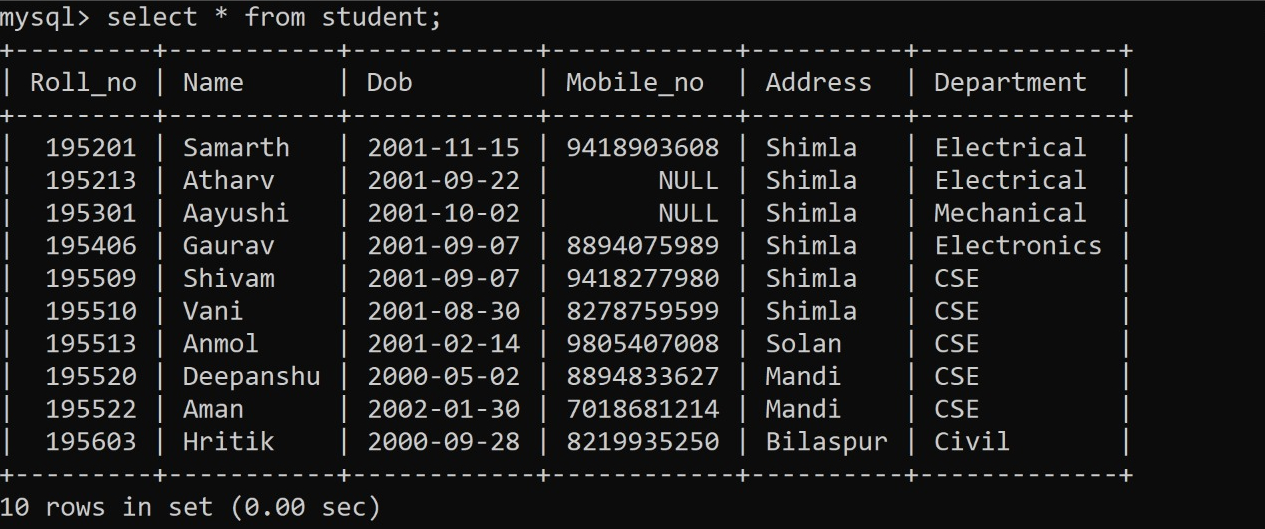
4. Inserting Data:

Data is inserted into table as a row using INSERT INTO statement.

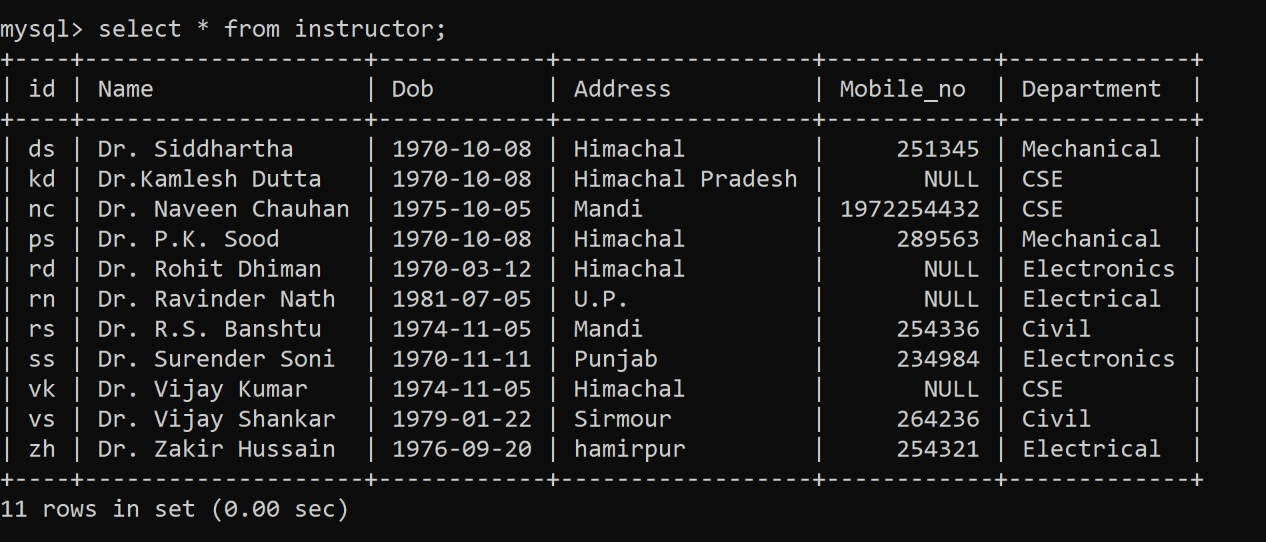
* + INSERTING DATA IN **DEPARTMENT** TABLE:
  + INSERT INTO DEPARTMENT VALUES ("CSE", "Dr.Teek Sharma", 300, 40);
  + INSERT INTO DEPARTMENT VALUES ("Civil", "Dr. R.S. Banshtu", 400, 30);
  + INSERT INTO DEPARTMENT VALUES ("Electrical", "Dr. Raj Jarial", 300, 35);
  + INSERT INTO DEPARTMENT VALUES ("Electronics", "Er. K.S. Pandey", 200, 30);
  + INSERT INTO DEPARTMENT VALUES ("Mechanical", "Dr.Ram Chauhan", 250, 35);
  + INSERT INTO DEPARTMENT VALUES ("Chemistry", "Dr. Pamita Awasth", 100, 35);
  + INSERT INTO DEPARTMENT VALUES ("Architecture", "Dr. I.P. Singh", 500, 25);



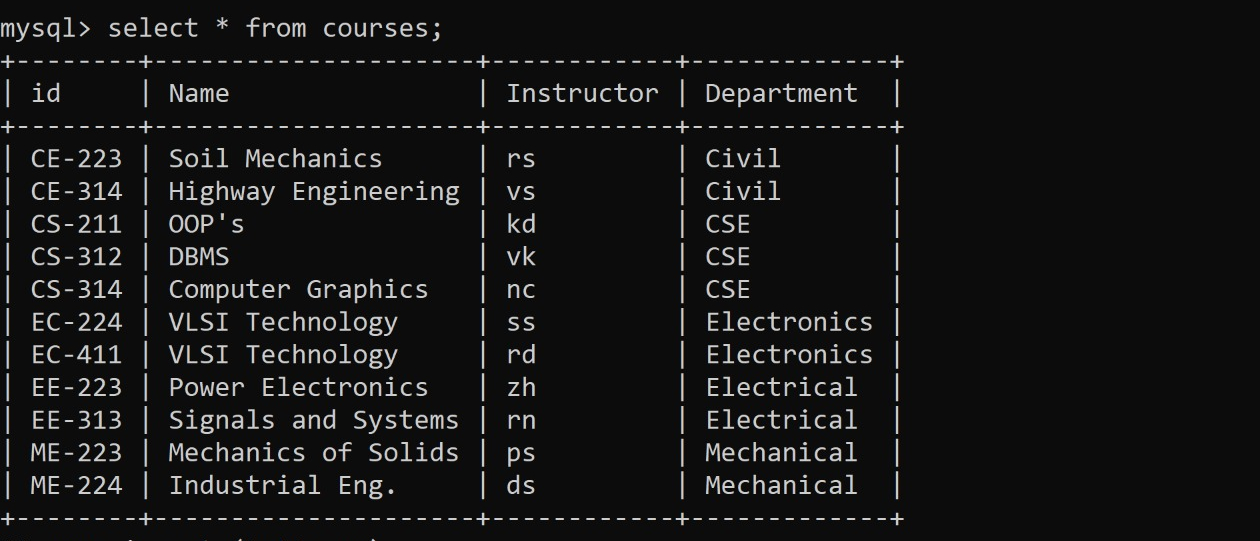
* + INSERTING DATA IN **STUDENT** TABLE:
  + INSERT INTO STUDENT VALUES (195509, "Shivam", "2001-09-07", 9418277980, "Shimla", "CSE");
  + INSERT INTO STUDENT VALUES (195510, "Vani", "2001-08-30", 8278759599, "Shimla", "CSE");
  + INSERT INTO STUDENT VALUES (195513, "Anmol", "2001-02-14", 9805407008, "Solan", "CSE");
  + INSERT INTO STUDENT VALUES (195520, "Deepanshu", "2000-05-02", 8894833627, "Mandi", "CSE");
  + INSERT INTO STUDENT VALUES (195522, "Aman", "2002-01-30", 7018681214, "Mandi", "CSE");
  + INSERT INTO STUDENT VALUES (195201, "Samarth", "2001-11-15", 9418903608, "Shimla", "Electrical");
  + INSERT INTO STUDENT VALUES (195603, "Hritik", "2000-09-28", 8219935250, "Bilaspur", "Civil");
  + INSERT INTO STUDENT VALUES(195406, "Gaurav", "2001-09-07", 8894075989, "Shimla", "Electronics");
  + INSERT INTO STUDENT VALUES (195301, "Aayushi", "2001-10-02", null, "Shimla", "Mechanical");
  + INSERT INTO STUDENT VALUES (195213, "Atharv", "2001-09-22", null, "Shimla", "Electrical");



* + INSERTING DATA IN **INSTRUCTOR** TABLE:
  + INSERT INTO INSTRUCTOR VALUES ("kd", "Dr.Kamlesh Dutta", "1970-10-08", "Himachal Pradesh", null, "CSE");
  + INSERT INTO INSTRUCTOR VALUES ("nc", "Dr. Naveen Chauhan", "1970-10-08", "Mandi", 1972254432, "CSE");
  + INSERT INTO INSTRUCTOR VALUES ("vk", "Dr. Vijay Kumar", "1970-10-08", "Himachal", null, "CSE");
  + INSERT INTO INSTRUCTOR VALUES ("rs", "Dr. R.S. Banshtu", "1974-11-05", "Mandi", 254336, "Civil");
  + INSERT INTO INSTRUCTOR VALUES("vs", "Dr. Vijay Shankar", "1979-01-22", "Sirmour", 264236, "Civil");
  + INSERT INTO INSTRUCTOR VALUES ("ss", "Dr. Surender Soni", "1970-11-11", "Punjab", 234984, "Electronics");
  + INSERT INTO INSTRUCTOR VALUES ("rd", "Dr. Rohit Dhiman", "1970-03-12", "Himachal", null, "Electronics");
  + INSERT INTO INSTRUCTOR VALUES ("zh", "Dr. Zakir Hussain", "1976-09-20", "Hamirpur", 254321, "Electrical");
  + INSERT INTO INSTRUCTOR VALUES ("rn", "Dr. Ravinder Nath", "1981-07-05", "U.P.", null, "Electrical");
  + INSERT INTO INSTRUCTOR VALUES ("ps", "Dr. P.K. Sood", "1970-10-08", "Himachal", 289563, "Mechanical");
  + INSERT INTO INSTRUCTOR VALUES ("ds", "Dr. Siddhartha", "1970-10-08", "Himachal", 251345, "Mechanical");



* + INSERTING DATA IN **COURSES** TABLE:
  + INSERT INTO COURSES VALUES ("CS-211", "OOP's", "kd", "CSE");
  + INSERT INTO COURSES VALUES ("CS-314", "Computer Graphics", "nc", "CSE");
  + INSERT INTO COURSES VALUES ("CS-312", "DBMS", "vk", "CSE");
  + INSERT INTO COURSES VALUES("CE-223","Soil Mechanics","rs","Civil");
  + INSERT INTO COURSES VALUES ("CE-314", "Highway Engineering", "vs", "Civil");
  + INSERT INTO COURSES VALUES ("EC-224", "VLSI Technology", "ss", "Electronics");
  + INSERT INTO COURSES VALUES ("EC-411", "VLSI Technology", "rd", "Electronics");
  + INSERT INTO COURSES VALUES ("EE-223", "Power Electronics", "zh", "Electrical");
  + INSERT INTO COURSES VALUES ("EE-313", "Signals and Systems", "rn", "Electrical");
  + INSERT INTO COURSES VALUES ("ME-223", "Mechanics of Solids", "ps", "Mechanical");
  + INSERT INTO COURSES VALUES ("ME-224", "Industrial Eng.", "ds", "Mechanical");



* + INSERTING DATA IN **TEACHES** TABLE:
  + INSERT INTO TEACHES VALUES (195509, "kd");
  + INSERT INTO TEACHES VALUES (195509, "nc");
  + INSERT INTO TEACHES VALUES (195509, "vk");
  + INSERT INTO TEACHES VALUES (195510, "kd");
  + INSERT INTO TEACHES VALUES (195510, "nc");
  + INSERT INTO TEACHES VALUES (195510, "vk");
  + INSERT INTO TEACHES VALUES (195513, "kd");
  + INSERT INTO TEACHES VALUES (195513, "nc");
  + INSERT INTO TEACHES VALUES (195513, "vk");
  + INSERT INTO TEACHES VALUES (195520, "kd");
  + INSERT INTO TEACHES VALUES (195520, "nc");
  + INSERT INTO TEACHES VALUES (195520, "vk");
  + INSERT INTO TEACHES VALUES (195522, "kd");
  + INSERT INTO TEACHES VALUES (195522, "nc");
  + INSERT INTO TEACHES VALUES (195522, "vk");
  + INSERT INTO TEACHES VALUES (195201, "zh");
  + INSERT INTO TEACHES VALUES (195201, "rn");
  + INSERT INTO TEACHES VALUES (195213, "zh");
  + INSERT INTO TEACHES VALUES (195213, "rn");
  + INSERT INTO TEACHES VALUES (195603, "rs");
  + INSERT INTO TEACHES VALUES (195603, "vs");
  + INSERT INTO TEACHES VALUES (195406, "ss");
  + INSERT INTO TEACHES VALUES (195406, "rd");
  + INSERT INTO TEACHES VALUES (195301, "ps");
  + INSERT INTO TEACHES VALUES (195301, "ds");



* + INSERTING DATA IN **ATTENDS** TABLE:
  + INSERT INTO ATTENDS VALUES (195509, "CS-314");
  + INSERT INTO ATTENDS VALUES (195509, "CS-312");
  + INSERT INTO ATTENDS VALUES (195510, "CS-211");
  + INSERT INTO ATTENDS VALUES (195510, "CS-314");
  + INSERT INTO ATTENDS VALUES (195510, "CS-312");
  + INSERT INTO ATTENDS VALUES (195513, "CS-211");
  + INSERT INTO ATTENDS VALUES (195513, "CS-314");
  + INSERT INTO ATTENDS VALUES (195513, "CS-312");
  + INSERT INTO ATTENDS VALUES (195520, "CS-211");
  + INSERT INTO ATTENDS VALUES (195520, "CS-314");
  + INSERT INTO ATTENDS VALUES (195520, "CS-312");
  + INSERT INTO ATTENDS VALUES (195522, "CS-211");
  + INSERT INTO ATTENDS VALUES (195522, "CS-314");
  + INSERT INTO ATTENDS VALUES (195522, "CS-312");
  + INSERT INTO ATTENDS VALUES (195201, "EE-223");
  + INSERT INTO ATTENDS VALUES (195201, "EE-313");
  + INSERT INTO ATTENDS VALUES (195213, "EE-223");
  + INSERT INTO ATTENDS VALUES (195213, "EE-313");
  + INSERT INTO ATTENDS VALUES (195603, "CE-223");
  + INSERT INTO ATTENDS VALUES (195603, "CE-314");
  + INSERT INTO ATTENDS VALUES (195406, "EC-224");
  + INSERT INTO ATTENDS VALUES (195406, "EC-411");
  + INSERT INTO ATTENDS VALUES (195301, "ME-223");
  + INSERT INTO ATTENDS VALUES (195301, "ME-224");

