UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

iAcademy
Quality Education Improving Lives

HERALD COLLEGE
KATHMANDU

| Academic Year | Module | Assessment Number | Assessment Type |
|---|---|---|---|
| S20 | DipIT07 - Introduction To Database System | A1 | Individual Report |

# School Management System

| | |
|---|---|
| Student Id | : NP03A190103 |
| Student Name | : Prashant Phuyal |
| Section | : DC06 |
| Module Leader | : Deepson Shrestha |
| Lecturer | : Deepson Shrestha |
| Submitted on | : 29/05/2020 |

## Acknowledgement

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

The success and final outcome of this project required a lot of guidance and assistance and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Mr. Deepson Shrestha, our module Leader for providing me an opportunity to do the report work and giving me all support and guidance. Also, I want to thank Mr. Sachit Tandukar and Mr. Subiran Shrestha Sir who helped me complete the project. I am extremely thankful to Herald College and teachers for providing such a great opportunity, support, guidance and providing all the necessary information of Course Work.

# Table of Contents

# I. System Description

## A. MySQL

MySQL is an open source relational database management system (RDBMS) with a client-server model. RDBMS is a software or service used to create and manage databases based on a relational model. A database is an organized collection of various forms of data. It is also known as a structured set of data that is accessible in many ways through computer.

The aim of this case study is to design and develop a database for the school to manage the record of students, classes, courses, subjects, teachers and books in the school. It also maintain the records of students assigned in class, staff departments, members of library, subject in courses, examination given by students, subject marks etc. in MySQL database.

## B. Database Description

School management system is an information system to manage school related data and day to day activities. In school, there are students, teachers, Courses, Staff, library, Departments, classes and subjects etc. There are courses where students enrolls, records of parent identified by students, classes for courses, attendance of students of each classes, subjects inside each courses which is taught by specific teachers, teachers categorized to specific department as per their subjects, records of staffs in school who works in different fields, teaching hours of each subjects per day, Examination which is to be held per year and marks scored by students in each subjects. Also, there is a Library in school where there are many books which can be issued by the members of library, members can either issue zero or more than one books as per their choice.

All the records are store in their own specific tables, for example there is a table for Members and books where record of members are kept in members table and record of books are kept in Books table respectively and to relate them together there is a table named, issued books through which library manager will able to know which books are issued by which members easily.

In this way, data can be secured, organized, managed and accessed whenever it is required which in turn improve the reliability of schools information.

## II. Data Dictionaries

Following are the tables along with constraints used in School Management System database.

### A. School Table

This table consists of information of school name, its address and CST No. of school.

*Constraints*: Data can't be empty in this table, and there is no foreign key and primary key. BIGINT is used so that large number can be added in the table.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| school_name | VARCHAR | 60 | NOT NULL | Name of school |
| school_address | VARCHAR | 60 | NOT NULL | School address |
| CST No. | BIGINT | - | NOT NULL | School SCT No. |

### B. Student Table

This table consists of all the necessary information of the students like student name, date of birth, Student roll number, student id and gender.

*Constraints*: Student id is used as primary key and Class id is used as Foreign key which so that class table references student table for classes.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| student_id | VARCHAR | 10 | PK | Student id |
| student_name | VARCHAR | 30 | NOT NULL | Name of Student |
| Class_id | VARCHAR | 10 | FK | ID of class where student study |
| roll_no | INT | 11 | NOT NULL | Student Roll No. |
| DOB | DATE | - | NOT NULL | Date of Birth |
| gender | VARCHAR | 20 | NOT NULL | Student Gender |

### C. Parent Table

This table stores the detail of parents which consists attribute like parent name, phone number and parent address.

*Constraints*: Student ID is used as a foreign key to reference the student table and BIGINT is used to store large Integer character.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| parent_Name | VARCHAR | 30 | NOT NULL | Parent’s Name |
| student_id | VARCHAR | 10 | FK | Student Id |
| phone_number | BIGINT | - | NOT NULL | Parent’s Phone No. |
| parent_address | VARCHAR | 40 | NOT NULL | Parents Address |

### D. Course Table

This table stores the courses of education school provides and student can choose the courses as per their need.

*Constraints*: Course ID is primary key here and corresponding course id should be in subject table.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| course_id | VARCHAR | 10 | PK | Course ID |
| course_name | VARCHAR | 20 | NOT NULL | Name of Course |

### E. Attendance Table

This table stores the data about the total numbers of study days student were present

*Constraints*: This table consists of attribute as total_class and total_present and student _d is a foreign key to reference the table, student.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| student_id | VARCHAR | 10 | FK | Student ID |
| total_class | INT | 11 | NOT NULL | Total study class |
| total_present | INT | 11 | NOT NULL | Total present days |

### F. Class Table

This table stores the total number of classes in the school and student table reference this table to keep the record of the student’s class

*Constraints*: This table consists of attribute as class id as a primary key and class _name student are associated with.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| class_id | VARCHAR | 10 | PK | Class ID |
| class_name | VARCHAR | 20 | NOT NULL | Name of Class |

### G. Subject Table

Subject table holds the data of the subject as per the courses and classes.

*Constraints*: This table consists of attribute as subject id as a primary key and subject name as name of the subject. Class id and course id references class table and course table and relation between them.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| subject_id | VARCHAR | 10 | PK | Subject ID |
| subject_name | VARCHAR | 20 | NOT NULL | Subject Name |
| class_id | VARCHAR | 10 | FK | Class ID |
| course_id | VARCHAR | 10 | FK | Course ID |

### H. Period Hour Table

This table consists of information related the time period of the each subject.

*Constraints*: This table has attribute as subject id as a foreign key which references subject table in order to know each subsect marks. FLOAT is used to store decimal value.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| subject_id | VARCHAR | 10 | FK | Subject ID |
| time_period | FLOAT | - | NOT NULL | Time in hour |

### I. Teacher Table

This table consists of all the necessary information of the teachers like teacher name, teacher address.

*Constraints*: Teacher id is used as primary key to uniquely identity the teacher and Department id is used as foreign key to find which department teachers is associated with.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| teacher_id | INT | 11 | PK | Teacher ID |
| teacher_name | VARCHAR | 30 | NOT NULL | Teacher Name |
| teacher_address | VARCHAR | 40 | NOT NULL | Teacher Address |
| department_id | VARCHAR | 10 | FK | Teacher belonging Department |

### J. Department Table

This table holds the data for department name and department id.

*Constraints*: Here, department id is primary key which is to be referenced on Teacher and Staff table respectively.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| department_id | VARCHAR | 10 | PK | Department ID |
| department_name | VARCHAR | 50 | NOT NULL | Department Name |

### K. Marks Table

In this table marks of each subject and their respective subject is recorded.

*Constraints*: This table have two foreign key i.e. student id and subject id referenced by student table and subject table respectively and attribute sub marks holds the marks of subject.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| student_id | VARCHAR | 10 | FK | Student ID |
| subject_id | VARCHAR | 10 | FK | Subject ID |
| sub_marks | INT | 11 | NOT NULL | Marks of each subject of students |

### L. Exam Table

This table consists of data regarding the exam and the date in which subject exam has to be executed.

*Constraints*: This table has primary key exam id which identifies this table and which is to be referenced in exam_subject table for relation between subject table and exam table.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| exam_id | VARCHAR | 10 | PK | Exam ID |
| exam_date | DATE | - | NOT NULL | Date of exam for each subjects |

### M. Staff Table

This table consists of all the necessary information of the Staff like staff name, work role staff has to do, Staff phone number and department id where staff is associated.

Constraints: Here staff id is primary key which uniquely identifies records and department id is foreign key which references department table.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| staff_id | VARCHAR | 10 | PK | Exam ID |
| staff_name | VARCHAR | 30 | NOT NULL | Staff Name |

| | | | | |
|---|---|---|---|---|
| work_role | VARCHAR | 40 | NOT NULL | Task staff need to do |
| staff_phone | BIGINT | - | NOT NULL | Staff Phone number |
| department_id | VARCHAR | 10 | FK | Department ID |

### N. Library Table

This table stores information of library, its name and id.

*Constraints*: This table has library id as primary key, it's not referenced in any table but added incase student as more than one library.

| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
|---|---|---|---|---|
| library_id | INT | 11 | PK | Library ID |
| library_name | VARCHAR | 40 | NOT NULL | Library Name |

### O. Books Table

The books present in the library is added or inserted into this table, which stores the name of book and writer's name.

*Constraints*: There is auto incremented book id which starts from 1000 and uniquely identifies record of books, book id is used as foreign key in table (issued books) as a relation between member table and books table.

| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
|---|---|---|---|---|
| book_id | INT | 11 | PK | Book ID, Auto Increment |
| book_name | VARCHAR | 40 | NOT NULL | Book Name |
| writer | VARCHAR | 30 | NOT NULL | Book Writer Name |

### P. Members Table

This table consists of member id and student id

*Constraints*: student id as a foreign key works to provide details of the members referenced by student table. Primary key member id is used as foreign key to reference members table in issued books table.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| member_id | VARCHAR | 10 | PK | Member ID |
| student_id | VARCHAR | 10 | FK | Student ID |

### Q. issued_books table

This table issued books is a relation between two table members and book.

*Constraints*: Two foreign key of member id and book id references member and book table respectively, null value is not accepted.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| member_id | VARCHAR | 10 | FK | Member ID |
| book_id | INT | 11 | FK | Book ID |

### R. student_course Table

This table student_course is a relation between two table student and course.

*Constraints*: Two foreign key of course id and student id references course and student table respectively, null value is not accepted.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| course_id | VARCHAR | 10 | FK | Course ID |
| student_id | VARCHAR | 10 | FK | Student ID |

### S. exam_subject Table

This table exam_subject is a relation between two table exam and table.

*Constraints*: Two foreign key of exam id and subject id references exam and subject table respectively, null value is not accepted.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| exam_id | VARCHAR | 10 | FK | Exam ID |
| subject_id | VARCHAR | 10 | FK | Subject ID |

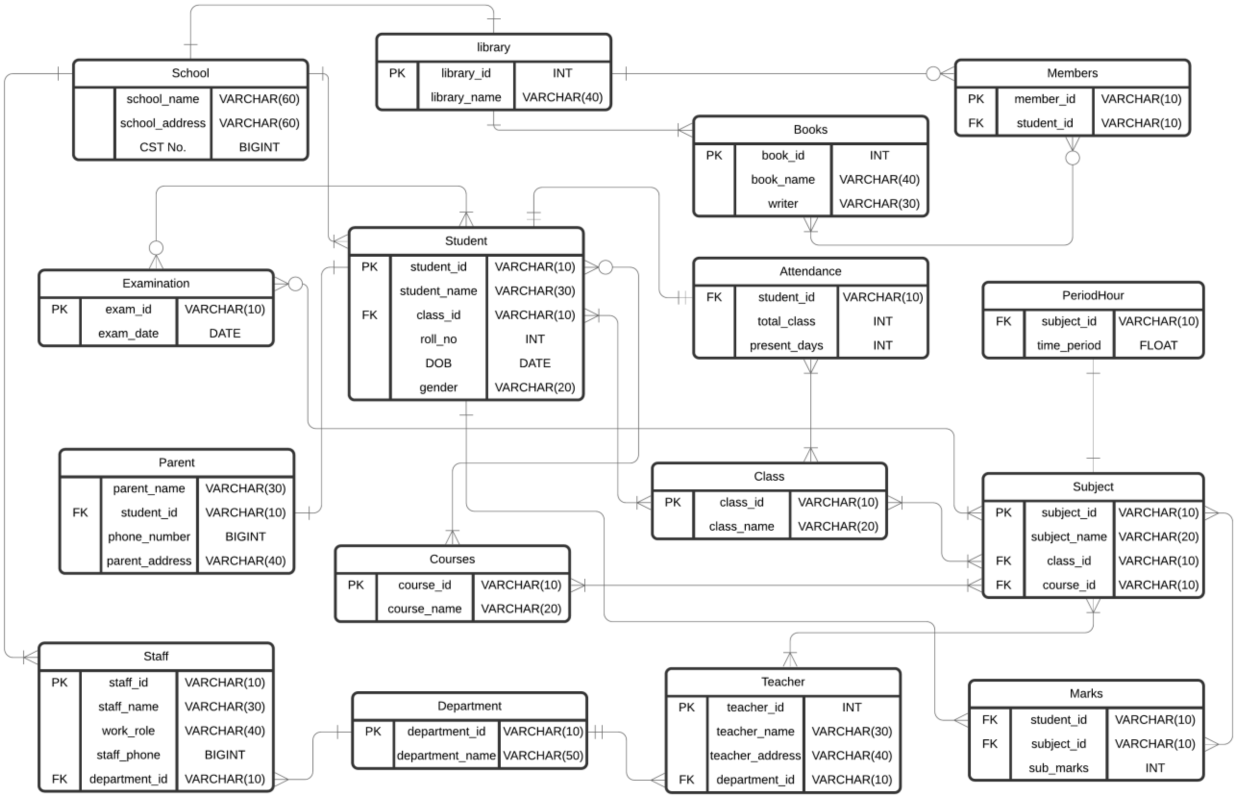
### T. teacher_teaches Table

This table teacher_teaches is a relation between two table teacher and subject.

*Constraints*: Two foreign key of teacher id and subject id references teacher and subject table respectively, null value is not accepted.

| Field Name | Data Type | Field Length | Constraint | Description |
|---|---|---|---|---|
| teacher_id | INT | 11 | FK | Teacher ID |
| subject_id | VARCHAR | 10 | FK | Subject ID |

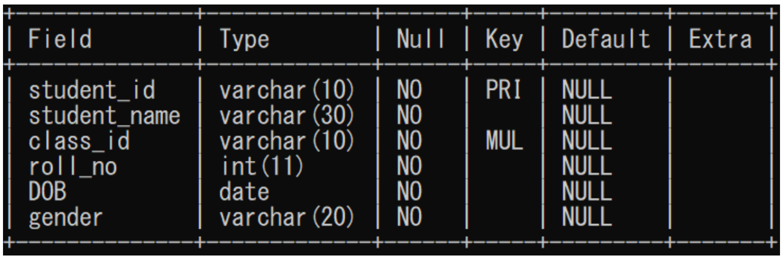
## III. ER Diagram



# IV. Create Databases and Tables / Insert sufficient Data

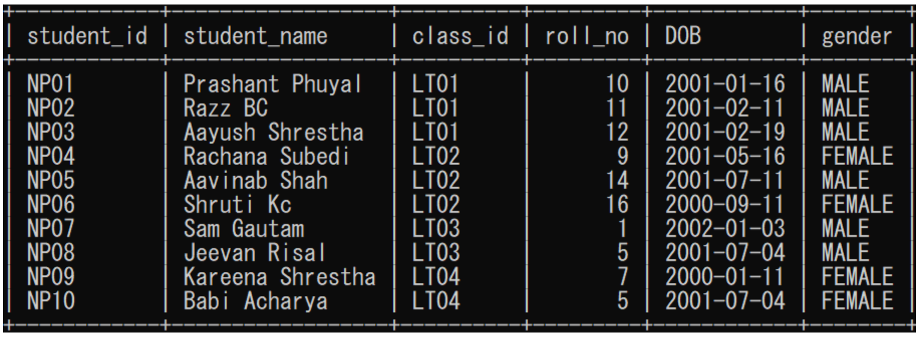
## A. Student Table

```
CREATE TABLE `Student` (
  `student_id` VARCHAR(10),
  `student_name` VARCHAR(30) NOT NULL,
  `class_id` VARCHAR(10) NOT NULL,
  `roll_no` INT NOT NULL,
  `DOB` DATE NOT NULL,
  `gender` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`student_id`)
  FOREIGN KEY (`class_id`) REFERENCES Class(`class_id`);
);
```



```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| student_id   | varchar(10) | NO   | PRI | NULL    |       |
| student_name | varchar(30) | NO   |     | NULL    |       |
| class_id     | varchar(10) | NO   | MUL | NULL    |       |
| roll_no      | int(11)     | NO   |     | NULL    |       |
| DOB          | date        | NO   |     | NULL    |       |
| gender       | varchar(20) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

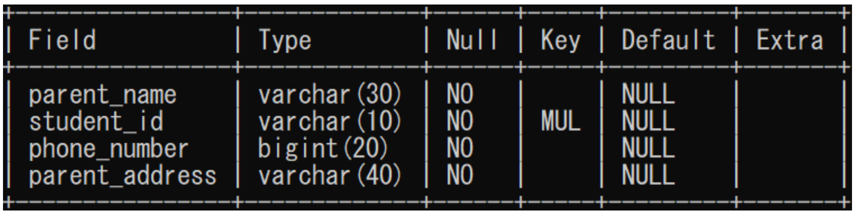
```
INSERT INTO `Student` VALUES
('NP01', 'Prashant Phuyal', 'LT01', '10', '2001-01-16', 'MALE'),
('NP02', 'Razz BC', 'LT01', '11', '2001-02-11', 'MALE'),
('NP03', 'Aayush Shrestha', 'LT01', '12', '2001-02-19', 'MALE'),
('NP04', 'Rachana Subedi', 'LT02', '09', '2001-05-16', 'FEMALE'),
('NP05', 'Aavinab Shah', 'LT02', '14', '2001-07-11', 'MALE'),
('NP06', 'Shruti Kc', 'LT02', '16', '2000-09-11', 'FEMALE'),
('NP07', 'Sam Gautam', 'LT03', '01', '2002-01-03', 'MALE'),
('NP08', 'Jeevan Risal', 'LT03', '05', '2001-07-04', 'MALE'),
('NP09', 'Kareena Shrestha', 'LT04', '07', '2000-01-11', 'FEMALE'),
('NP10', 'Babi Acharya', 'LT04', '05', '2001-07-04', 'FEMALE');
```



```
+------------+------------------+----------+---------+------------+--------+
| student_id | student_name     | class_id | roll_no | DOB        | gender |
+------------+------------------+----------+---------+------------+--------+
| NP01       | Prashant Phuyal  | LT01     |      10 | 2001-01-16 | MALE   |
| NP02       | Razz BC          | LT01     |      11 | 2001-02-11 | MALE   |
| NP03       | Aayush Shrestha  | LT01     |      12 | 2001-02-19 | MALE   |
| NP04       | Rachana Subedi   | LT02     |       9 | 2001-05-16 | FEMALE |
| NP05       | Aavinab Shah     | LT02     |      14 | 2001-07-11 | MALE   |
| NP06       | Shruti Kc        | LT02     |      16 | 2000-09-11 | FEMALE |
| NP07       | Sam Gautam       | LT03     |       1 | 2002-01-03 | MALE   |
| NP08       | Jeevan Risal     | LT03     |       5 | 2001-07-04 | MALE   |
| NP09       | Kareena Shrestha | LT04     |       7 | 2000-01-11 | FEMALE |
| NP10       | Babi Acharya     | LT04     |       5 | 2001-07-04 | FEMALE |
+------------+------------------+----------+---------+------------+--------+
```

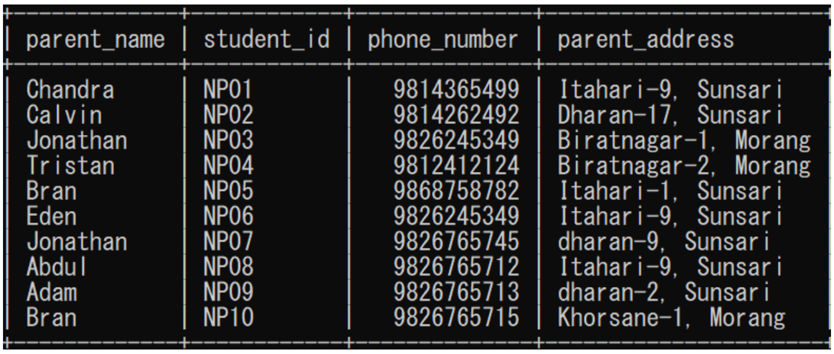
## B. Parent Table

```
CREATE TABLE `Parent` (
  `parent_name` VARCHAR(30) NOT NULL,
  `student_id` VARCHAR(10) NOT NULL,
  `phone_number` BIGINT NOT NULL,
  `parent_address` VARCHAR(40) NOT NULL,
  FOREIGN KEY (`student_id`) REFERENCES Student(`student_id`)
);
```



```
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| parent_name    | varchar(30) | NO   |     | NULL    |       |
| student_id     | varchar(10) | NO   | MUL | NULL    |       |
| phone_number   | bigint(20)  | NO   |     | NULL    |       |
| parent_address | varchar(40) | NO   |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

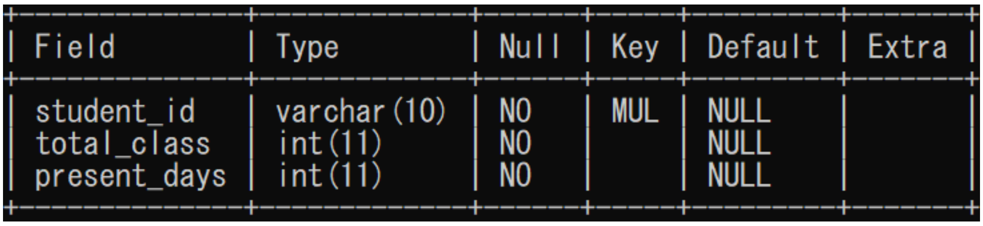
```
INSERT INTO `Parent` VALUES
('Chandra', 'NP01', '9814365499', 'Itahari-9, Sunsari'),
('Calvin', 'NP02', '9814262492', 'Dharan-17, Sunsari'),
('Jonathan', 'NP03', '9826245349', 'Biratnagar-1, Morang'),
('Tristan', 'NP04', '9812412124', 'Biratnagar-2, Morang'),
('Bran', 'NP05', '9868758782', 'Itahari-1, Sunsari'),
('Eden', 'NP06', '9826245349', 'Itahari-9, Sunsari'),
('Jonathan', 'NP07', '9826765745', 'dharan-9, Sunsari'),
('Abdul', 'NP08', '9826765712', 'Itahari-9, Sunsari'),
('Adam', 'NP09', '9826765713', 'dharan-2, Sunsari'),
('Bran', 'NP10', '9826765715', 'Khorsane-1, Morang');
```



```
+-------------+------------+--------------+----------------------+
| parent_name | student_id | phone_number | parent_address       |
+-------------+------------+--------------+----------------------+
| Chandra     | NP01       |   9814365499 | Itahari-9, Sunsari   |
| Calvin      | NP02       |   9814262492 | Dharan-17, Sunsari   |
| Jonathan    | NP03       |   9826245349 | Biratnagar-1, Morang |
| Tristan     | NP04       |   9812412124 | Biratnagar-2, Morang |
| Bran        | NP05       |   9868758782 | Itahari-1, Sunsari   |
| Eden        | NP06       |   9826245349 | Itahari-9, Sunsari   |
| Jonathan    | NP07       |   9826765745 | dharan-9, Sunsari    |
| Abdul       | NP08       |   9826765712 | Itahari-9, Sunsari   |
| Adam        | NP09       |   9826765713 | dharan-2, Sunsari    |
| Bran        | NP10       |   9826765715 | Khorsane-1, Morang   |
+-------------+------------+--------------+----------------------+
```

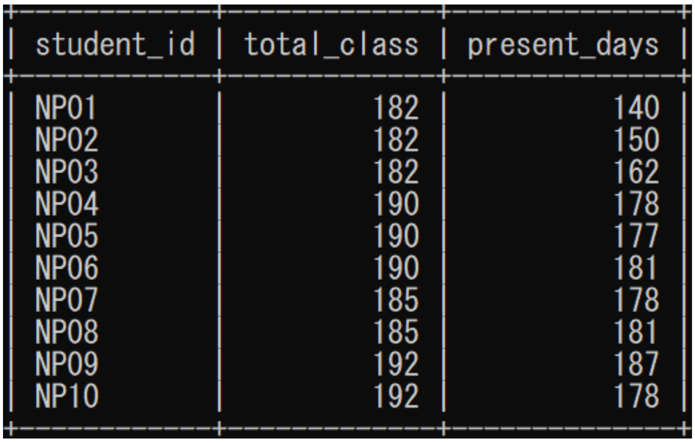
## C. Attandance Table

```
CREATE TABLE `Attendance` (
  `student_id` VARCHAR(10) NOT NULL,
  `total_class` INT NOT NULL,
  `present_days` INT NOT NULL,
  FOREIGN KEY (`student_id`) REFERENCES Student(`student_id`)
);
```



```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| student_id   | varchar(10) | NO   | MUL | NULL    |       |
| total_class  | int(11)     | NO   |     | NULL    |       |
| present_days | int(11)     | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

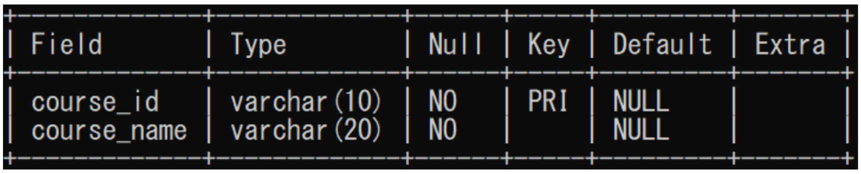
```
INSERT INTO `Attendance` VALUES
('NP01', '182', '140'), ('NP02', '182','150'),
('NP03', '182','162'), ('NP04', '190','178'),
('NP05', '190','177'), ('NP06', '190','181'),
('NP07', '185','178'), ('NP08', '185','181'),
('NP09', '192','187'), ('NP10', '192','178');
```



```
+------------+-------------+--------------+
| student_id | total_class | present_days |
+------------+-------------+--------------+
| NP01       |         182 |          140 |
| NP02       |         182 |          150 |
| NP03       |         182 |          162 |
| NP04       |         190 |          178 |
| NP05       |         190 |          177 |
| NP06       |         190 |          181 |
| NP07       |         185 |          178 |
| NP08       |         185 |          181 |
| NP09       |         192 |          187 |
| NP10       |         192 |          178 |
+------------+-------------+--------------+
```

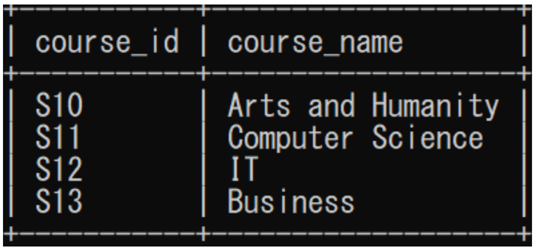
## D. Courses Table

```
CREATE TABLE `Courses` (
  `course_id` VARCHAR(10),
  `course_name` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`course_id`)
);
```



```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| course_id   | varchar(10) | NO   | PRI | NULL    |       |
| course_name | varchar(20) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

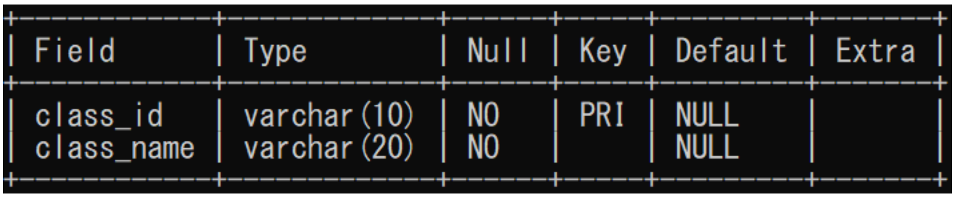
```
INSERT INTO `Courses` VALUES
('S10', 'Arts and Humanity'),
('S11', 'Computer Science'),
('S12', 'IT'),
('S13', 'Business');
```



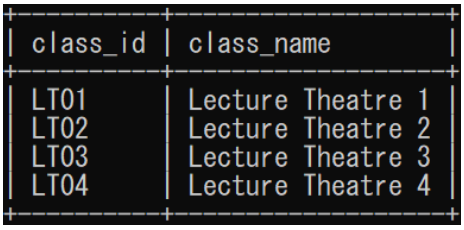
```
+-----------+-------------------+
| course_id | course_name       |
+-----------+-------------------+
| S10       | Arts and Humanity |
| S11       | Computer Science  |
| S12       | IT                |
| S13       | Business          |
+-----------+-------------------+
```

## E. Class Table

```
CREATE TABLE `Class` (
  `class_id` VARCHAR(10),
  `class_name` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`class_id`)
);
```



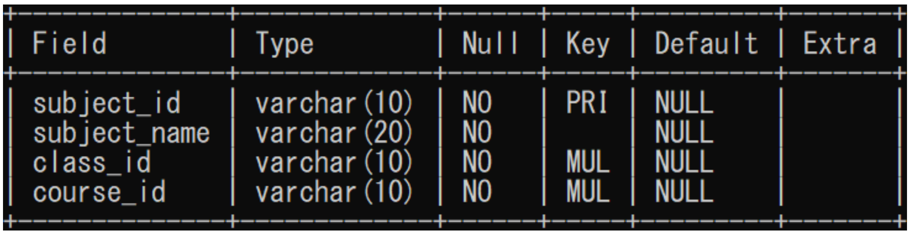
```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| class_id   | varchar(10) | NO   | PRI | NULL    |       |
| class_name | varchar(20) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
INSERT INTO `class` VALUES
('LT01','Lecture Theatre 1'),
('LT02','Lecture Theatre 2'),
('LT03','Lecture Theatre 3'),
('LT04','Lecture Theatre 4');
```



```
+----------+-------------------+
| class_id | class_name        |
+----------+-------------------+
| LT01     | Lecture Theatre 1 |
| LT02     | Lecture Theatre 2 |
| LT03     | Lecture Theatre 3 |
| LT04     | Lecture Theatre 4 |
+----------+-------------------+
```

## F. Subject Table

```
CREATE TABLE `Subject` (
  `subject_id` VARCHAR(10),
  `subject_name` VARCHAR(20) NOT NULL,
  `class_id` VARCHAR(10) NOT NULL,
  `course_id` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`subject_id`),
  FOREIGN KEY (`class_id`) REFERENCES Class(`class_id`),
  FOREIGN KEY (`course_id`) REFERENCES Courses(`course_id`)
);
```



```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| subject_id   | varchar(10) | NO   | PRI | NULL    |       |
| subject_name | varchar(20) | NO   |     | NULL    |       |
| class_id     | varchar(10) | NO   | MUL | NULL    |       |
| course_id    | varchar(10) | NO   | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

```
INSERT INTO `Subject` VALUES
('E01', 'History', 'LT01','S10'),
('E02', 'Music and Art','LT01','S10'),
('E03', 'Philosophy', 'LT01','S10'),
('F01', 'Algorithms', 'LT02','S11'),
('F02', 'Software Development', 'LT02','S11'),
('F03', 'Computer Security', 'LT02','S11'),
('F04', 'Design and Product', 'LT02','S11'),
('H10', 'Networking', 'LT03','S12'),
('H11', 'Cloud Computing', 'LT03','S12'),
```

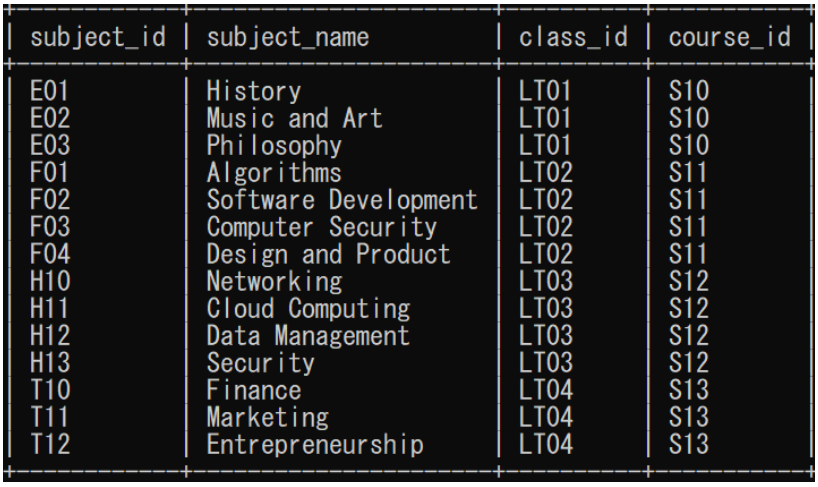
('H12', 'Data Management', 'LT03','S12'),

('H13', 'Security', 'LT03','S12'),

('T10', 'Finance', 'LT04','S13'),

('T11', 'Marketing', 'LT04','S13'),

('T12', 'Entrepreneurship', 'LT04','S13');



```
+------------+-----------------------+----------+-----------+
| subject_id | subject_name          | class_id | course_id |
+------------+-----------------------+----------+-----------+
| E01        | History               | LT01     | S10       |
| E02        | Music and Art         | LT01     | S10       |
| E03        | Philosophy            | LT01     | S10       |
| F01        | Algorithms            | LT02     | S11       |
| F02        | Software Development  | LT02     | S11       |
| F03        | Computer Security     | LT02     | S11       |
| F04        | Design and Product    | LT02     | S11       |
| H10        | Networking            | LT03     | S12       |
| H11        | Cloud Computing       | LT03     | S12       |
| H12        | Data Management       | LT03     | S12       |
| H13        | Security              | LT03     | S12       |
| T10        | Finance               | LT04     | S13       |
| T11        | Marketing             | LT04     | S13       |
| T12        | Entrepreneurship      | LT04     | S13       |
+------------+-----------------------+----------+-----------+
```

## G. PeriodHour Table

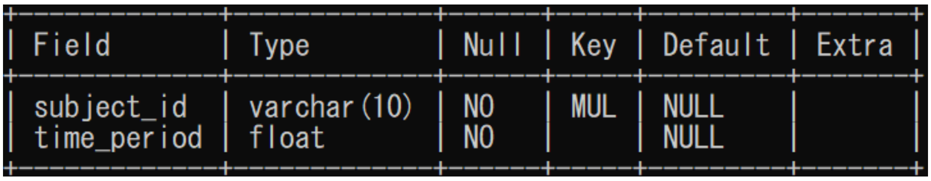
CREATE TABLE `PeriodHour` (

`subject_id` VARCHAR(10) NOT NULL,

`time_period` FLOAT NOT NULL,

FOREIGN KEY (`subject_id`) REFERENCES Subject(`subject_id`)

);



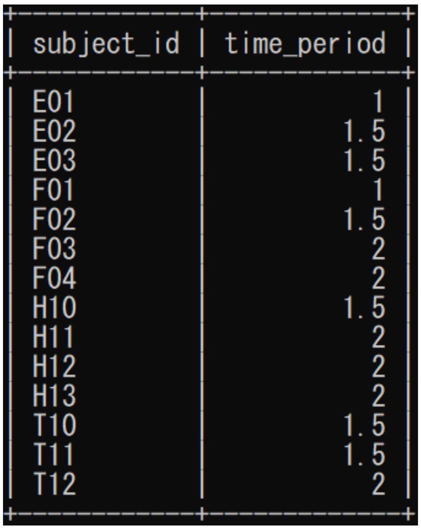
```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| subject_id  | varchar(10) | NO   | MUL | NULL    |       |
| time_period | float       | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

INSERT INTO `PeriodHour` VALUES

('E01', '1'), ('E02', '1.5'), ('E03','1.5'), ('F01', '1'), ('F02', '1.5'),

('F03', '2'), ('F04', '2'), ('H10', '1.5'), ('H11', '2'), ('H12', '2'),

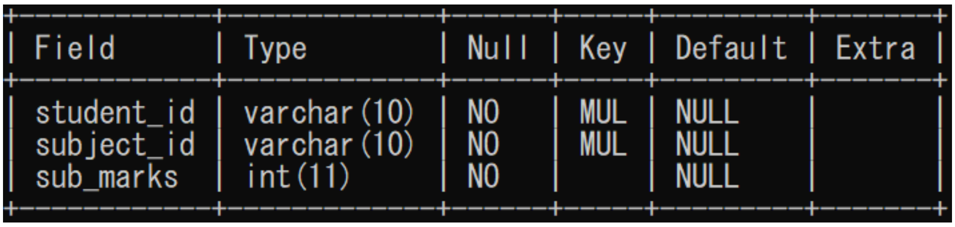
('H13', '2'), ('T10', '1.5'), ('T11','1.5'), ('T12', '2');



```
+------------+-------------+
| subject_id | time_period |
+------------+-------------+
| E01        |           1 |
| E02        |         1.5 |
| E03        |         1.5 |
| F01        |           1 |
| F02        |         1.5 |
| F03        |           2 |
| F04        |           2 |
| H10        |         1.5 |
| H11        |           2 |
| H12        |           2 |
| H13        |           2 |
| T10        |         1.5 |
| T11        |         1.5 |
| T12        |           2 |
+------------+-------------+
```

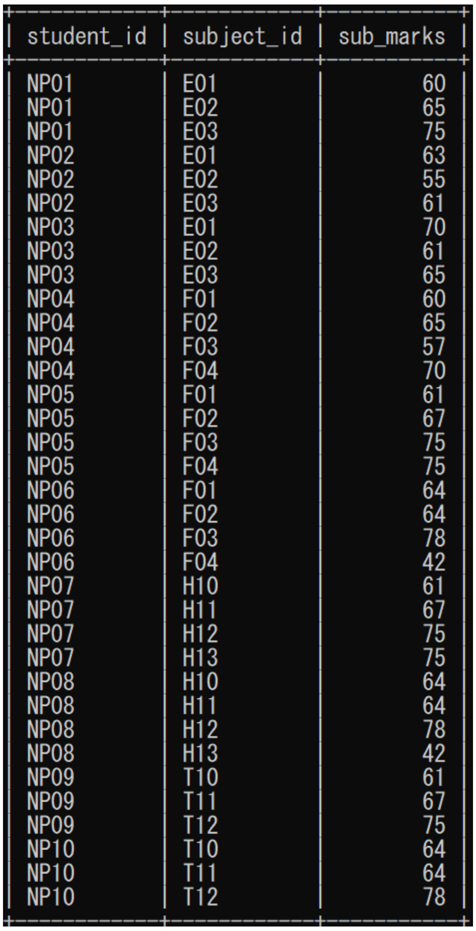
## H. Marks Table

```
CREATE TABLE `Marks` (
  `student_id` VARCHAR(10) NOT NULL,
  `subject_id` VARCHAR(10) NOT NULL,
  `sub_marks` INT NOT NULL,
  FOREIGN KEY (`student_id`) REFERENCES Student(`student_id`),
  FOREIGN KEY (`subject_id`) REFERENCES Subject(`subject_id`)
);
```



```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| student_id | varchar(10) | NO   | MUL | NULL    |       |
| subject_id | varchar(10) | NO   | MUL | NULL    |       |
| sub_marks  | int(11)     | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

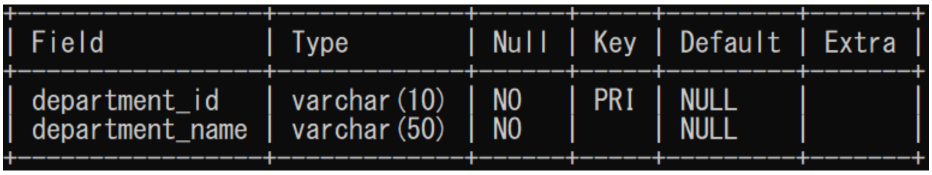
```
INSERT INTO `Marks` VALUES
('NP01','E01','60'), ('NP01','E02','65'), ('NP01','E03','75'),
('NP02','E01','63'), ('NP02','E02','55'), ('NP02','E03','61'),
('NP03','E01','70'), ('NP03','E02','61'), ('NP03','E03','65'),
('NP04','F01','60'), ('NP04','F02','65'), ('NP04','F03','57'), ('NP04','F04','70'),
('NP05','F01','61'), ('NP05','F02','67'), ('NP05','F03','75'), ('NP05','F04','75'),
('NP06','F01','64'), ('NP06','F02','64'), ('NP06','F03','78'), ('NP06','F04','42'),
('NP07','H10','61'), ('NP07','H11','67'), ('NP07','H12','75'), ('NP07','H13','75'),
('NP08','H10','64'), ('NP08','H11','64'), ('NP08','H12','78'), ('NP08','H13','42'),
('NP09','T10','61'), ('NP09','T11','67'), ('NP09','T12','75'),
('NP10','T10','64'), ('NP10','T11','64'), ('NP10','T12','78');
```



```
+------------+------------+-----------+
| student_id | subject_id | sub_marks |
+------------+------------+-----------+
| NP01       | E01        |        60 |
| NP01       | E02        |        65 |
| NP01       | E03        |        75 |
| NP02       | E01        |        63 |
| NP02       | E02        |        55 |
| NP02       | E03        |        61 |
| NP03       | E01        |        70 |
| NP03       | E02        |        61 |
| NP03       | E03        |        65 |
| NP04       | F01        |        60 |
| NP04       | F02        |        65 |
| NP04       | F03        |        57 |
| NP04       | F04        |        70 |
| NP05       | F01        |        61 |
| NP05       | F02        |        67 |
| NP05       | F03        |        75 |
| NP05       | F04        |        75 |
| NP06       | F01        |        64 |
| NP06       | F02        |        64 |
| NP06       | F03        |        78 |
| NP06       | F04        |        42 |
| NP07       | H10        |        61 |
| NP07       | H11        |        67 |
| NP07       | H12        |        75 |
| NP07       | H13        |        75 |
| NP08       | H10        |        64 |
| NP08       | H11        |        64 |
| NP08       | H12        |        78 |
| NP08       | H13        |        42 |
| NP09       | T10        |        61 |
| NP09       | T11        |        67 |
| NP09       | T12        |        75 |
| NP10       | T10        |        64 |
| NP10       | T11        |        64 |
| NP10       | T12        |        78 |
+------------+------------+-----------+
```

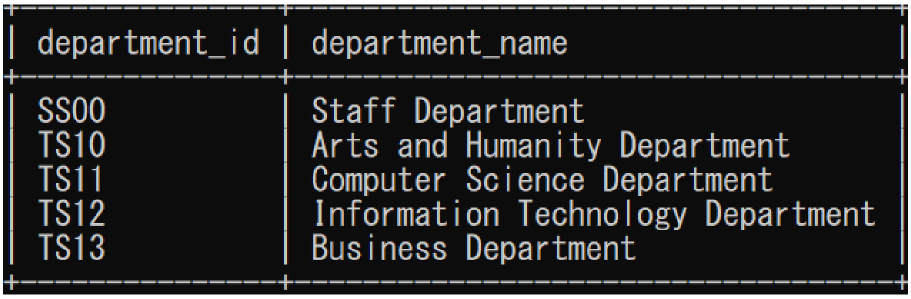
## I. Department Table

```
CREATE TABLE `Department` (
  `department_id` VARCHAR(10),
  `department_name` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`department_id`)
);
```



```
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| department_id   | varchar(10) | NO   | PRI | NULL    |       |
| department_name | varchar(50) | NO   |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
```

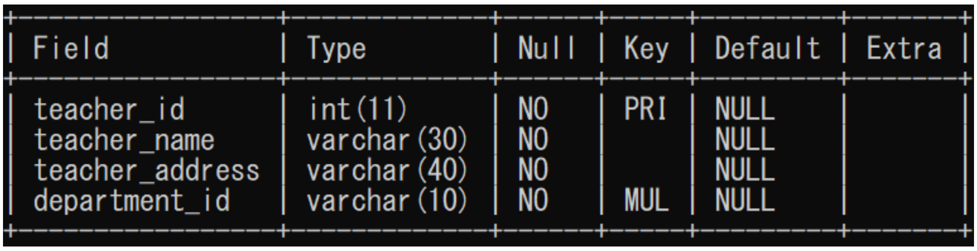
```
INSERT INTO `Department` VALUES
('TS10','Arts and Humanity Department'),
('TS11','Computer Science Department'),
('TS12','Information Technology Department'),
('TS13','Business Department'),
('SS00','Staff Department');
```



```
+---------------+-----------------------------------+
| department_id | department_name                   |
+---------------+-----------------------------------+
| SS00          | Staff Department                  |
| TS10          | Arts and Humanity Department      |
| TS11          | Computer Science Department       |
| TS12          | Information Technology Department |
| TS13          | Business Department               |
+---------------+-----------------------------------+
```

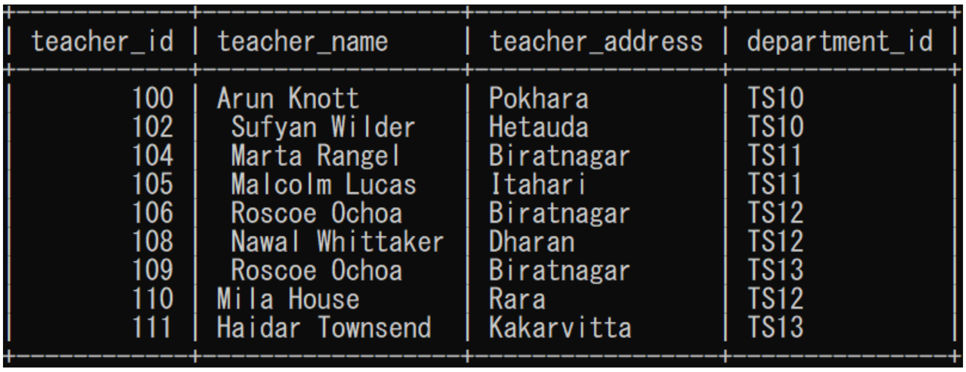
### J. Teacher Table

```
CREATE TABLE `Teacher` (
  `teacher_id` INT,
  `teacher_name` VARCHAR(30) NOT NULL,
  `teacher_address` VARCHAR(40) NOT NULL,
  `department_id` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`teacher_id`),
  FOREIGN KEY (`department_id`) REFERENCES Department(`department_id`)
);
```



```
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| teacher_id      | int(11)     | NO   | PRI | NULL    |       |
| teacher_name    | varchar(30) | NO   |     | NULL    |       |
| teacher_address | varchar(40) | NO   |     | NULL    |       |
| department_id   | varchar(10) | NO   | MUL | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
```

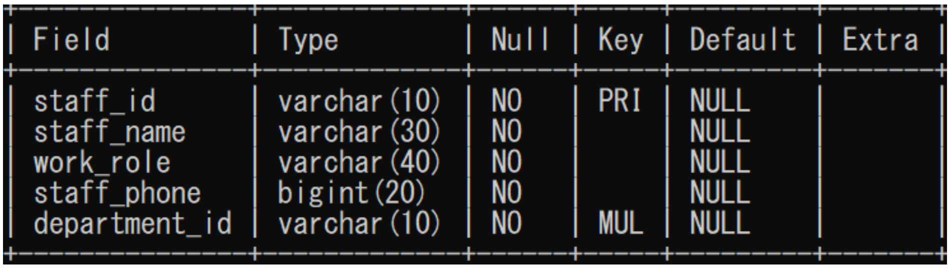
```
INSERT INTO `Teacher` VALUES
('100','Arun Knott','Pokhara','TS10'),
('102',' Sufyan Wilder', 'Hetauda', 'TS10'),
('104',' Marta Rangel','Biratnagar','TS11'),
('105',' Malcolm Lucas', 'Itahari', 'TS11'),
('106',' Roscoe Ochoa','Biratnagar','TS12'),
('108',' Nawal Whittaker', 'Dharan', 'TS12'),
('110', 'Mila House', 'Rara', 'TS12'),
('109',' Roscoe Ochoa','Biratnagar','TS13'),
('111','Haidar Townsend', 'Kakarvitta', 'TS13');
```



```
+------------+-----------------+-----------------+---------------+
| teacher_id | teacher_name    | teacher_address | department_id |
+------------+-----------------+-----------------+---------------+
|        100 | Arun Knott      | Pokhara         | TS10          |
|        102 |  Sufyan Wilder  | Hetauda         | TS10          |
|        104 |  Marta Rangel   | Biratnagar      | TS11          |
|        105 |  Malcolm Lucas  | Itahari         | TS11          |
|        106 |  Roscoe Ochoa   | Biratnagar      | TS12          |
|        108 |  Nawal Whittaker| Dharan          | TS12          |
|        109 |  Roscoe Ochoa   | Biratnagar      | TS13          |
|        110 | Mila House      | Rara            | TS12          |
|        111 | Haidar Townsend | Kakarvitta      | TS13          |
+------------+-----------------+-----------------+---------------+
```

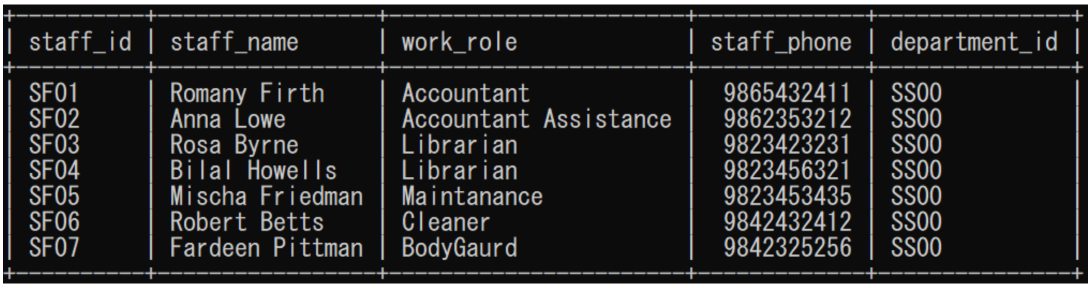
## K. Staff Table

```
CREATE TABLE `Staff` (
  `staff_id` VARCHAR(10),
  `staff_name` VARCHAR(30) NOT NULL,
  `work_role` VARCHAR(40) NOT NULL,
  `staff_phone` BIGINT NOT NULL,
  `department_id` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`staff_id`),
  FOREIGN KEY (`department_id`) REFERENCES Department(`department_id`)
);
```



```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| staff_id      | varchar(10) | NO   | PRI | NULL    |       |
| staff_name    | varchar(30) | NO   |     | NULL    |       |
| work_role     | varchar(40) | NO   |     | NULL    |       |
| staff_phone   | bigint(20)  | NO   |     | NULL    |       |
| department_id | varchar(10) | NO   | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

```
INSERT INTO `Staff` VALUES
('SF01','Romany Firth','Accountant','9865432411','SS00'),
('SF02','Anna Lowe', 'Accountant Assistance','9862353212','SS00'),
('SF03','Rosa Byrne','Librarian','9823423231','SS00'),
('SF04','Bilal Howells','Librarian','9823456321','SS00'),
('SF05','Mischa Friedman','Maintanance','9823453435','SS00'),
('SF06','Robert Betts','Cleaner','9842432412','SS00'),
('SF07','Fardeen Pittman','BodyGaurd','9842325256','SS00');
```



```
+----------+----------------+---------------------+-------------+---------------+
| staff_id | staff_name     | work_role           | staff_phone | department_id |
+----------+----------------+---------------------+-------------+---------------+
| SF01     | Romany Firth   | Accountant          |  9865432411 | SS00          |
| SF02     | Anna Lowe      | Accountant Assistance |  9862353212 | SS00          |
| SF03     | Rosa Byrne     | Librarian           |  9823423231 | SS00          |
| SF04     | Bilal Howells  | Librarian           |  9823456321 | SS00          |
| SF05     | Mischa Friedman | Maintanance         |  9823453435 | SS00          |
| SF06     | Robert Betts   | Cleaner             |  9842432412 | SS00          |
| SF07     | Fardeen Pittman | BodyGaurd           |  9842325256 | SS00          |
+----------+----------------+---------------------+-------------+---------------+
```

## L. Eximination Table

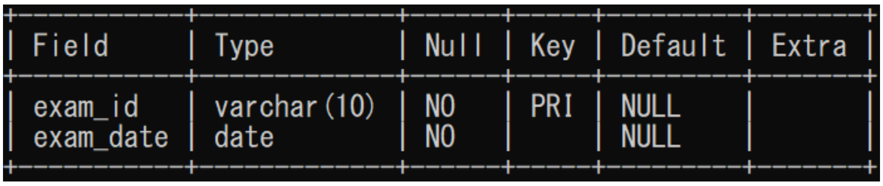
CREATE TABLE `Eximination` (

`exam_id` VARCHAR(10),

`exam_date` DATE NOT NULL,

PRIMARY KEY (`exam_id`)

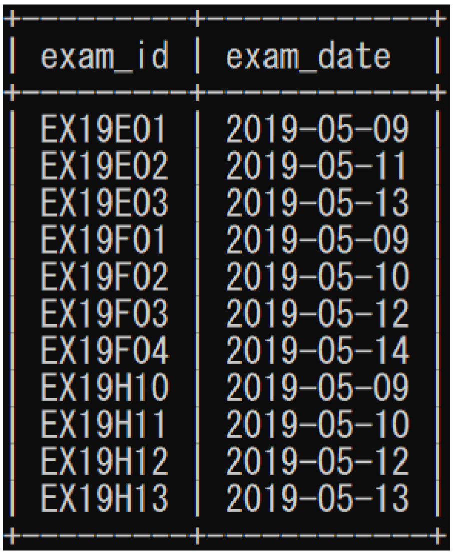
);



```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| exam_id   | varchar(10) | NO   | PRI | NULL    |       |
| exam_date | date        | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

INSERT INTO `Eximination` VALUES

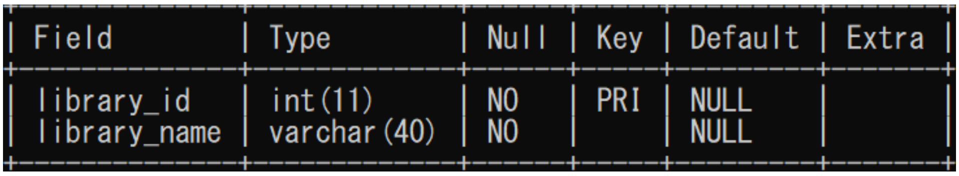
('EX19E01','2019-05-09'), ('EX19E02','2019-05-11'), ('EX19E03','2019-05-13'), ('EX19F01','2019-05-09'), ('EX19F02','2019-05-10'), ('EX19F03','2019-05-12'), ('EX19F04','2019-05-14'), ('EX19H10','2019-05-09'), ('EX19H11','2019-05-10'), ('EX19H12','2019-05-12'), ('EX19H13','2019-05-13');



```
+---------+------------+
| exam_id | exam_date  |
+---------+------------+
| EX19E01 | 2019-05-09 |
| EX19E02 | 2019-05-11 |
| EX19E03 | 2019-05-13 |
| EX19F01 | 2019-05-09 |
| EX19F02 | 2019-05-10 |
| EX19F03 | 2019-05-12 |
| EX19F04 | 2019-05-14 |
| EX19H10 | 2019-05-09 |
| EX19H11 | 2019-05-10 |
| EX19H12 | 2019-05-12 |
| EX19H13 | 2019-05-13 |
+---------+------------+
```

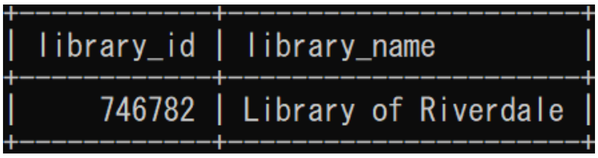
## M. Library Table

```
CREATE TABLE `Library` (
  `library_id` INT,
  `library_name` VARCHAR(40) NOT NULL,
  PRIMARY KEY (`library_id`)
);
```



```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| library_id   | int(11)     | NO   | PRI | NULL    |       |
| library_name | varchar(40) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

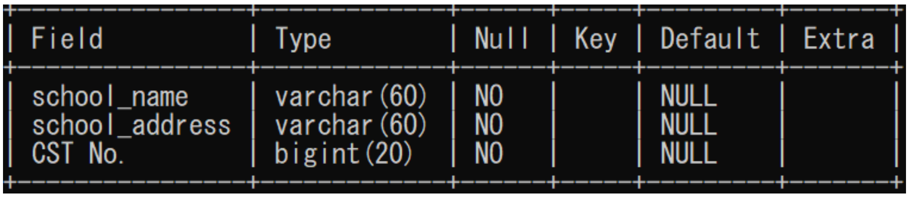
```
INSERT INTO `Library` VALUES
('746782','Library of Riverdale');
```



```
+------------+----------------------+
| library_id | library_name         |
+------------+----------------------+
|     746782 | Library of Riverdale |
+------------+----------------------+
```

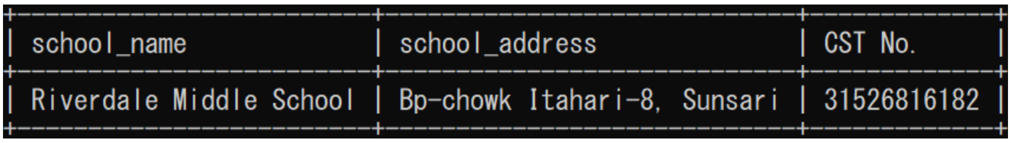
## N. School Table

```
CREATE TABLE `School` (
  `school_name` VARCHAR(60) NOT NULL,
  `school_address` VARCHAR(60) NOT NULL,
  `CST No.` BIGINT NOT NULL
);
```



```
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| school_name    | varchar(60) | NO   |     | NULL    |       |
| school_address | varchar(60) | NO   |     | NULL    |       |
| CST No.        | bigint(20)  | NO   |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

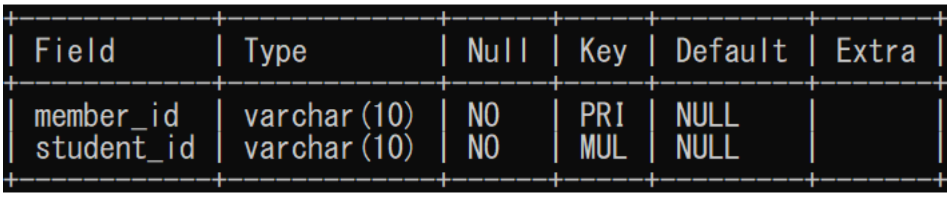
```
INSERT INTO `School` VALUES
('Riverdale Middle School', 'Bp-chowk Itahari-8, Sunsari', '31526816182');
```



```
+-------------------------+-----------------------------+-------------+
| school_name             | school_address              | CST No.     |
+-------------------------+-----------------------------+-------------+
| Riverdale Middle School | Bp-chowk Itahari-8, Sunsari | 31526816182 |
+-------------------------+-----------------------------+-------------+
```

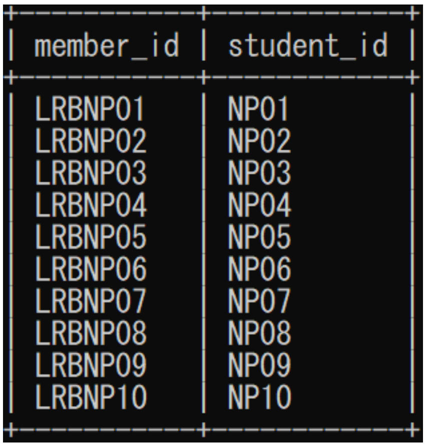
## O. Members Table

```
CREATE TABLE `Members` (
  `member_id` VARCHAR(10),
  `student_id` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`member_id`),
  FOREIGN KEY (`student_id`) REFERENCES Student(`student_id`)
);
```



```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| member_id  | varchar(10) | NO   | PRI | NULL    |       |
| student_id | varchar(10) | NO   | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

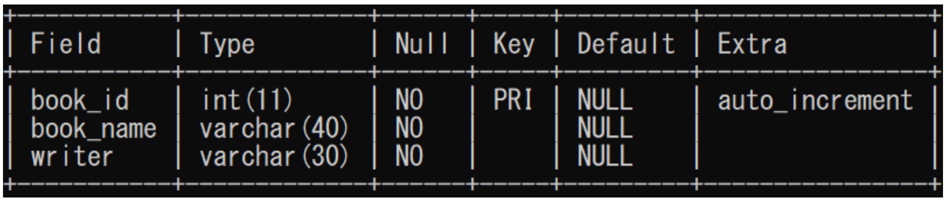
```
INSERT INTO `Members` VALUES
('LRBNP01','NP01'), ('LRBNP02','NP02'), ('LRBNP03','NP03'), ('LRBNP04','NP04'),
('LRBNP05','NP05'), ('LRBNP06','NP06'), ('LRBNP07','NP07'), ('LRBNP08','NP08'),
('LRBNP09','NP09'), ('LRBNP10','NP10');
```



```
+-----------+------------+
| member_id | student_id |
+-----------+------------+
| LRBNP01   | NP01       |
| LRBNP02   | NP02       |
| LRBNP03   | NP03       |
| LRBNP04   | NP04       |
| LRBNP05   | NP05       |
| LRBNP06   | NP06       |
| LRBNP07   | NP07       |
| LRBNP08   | NP08       |
| LRBNP09   | NP09       |
| LRBNP10   | NP10       |
+-----------+------------+
```

## P. Books Table

```
CREATE TABLE `Books` (
  `book_id` INT AUTO_INCREMENT,
  `book_name` VARCHAR(40) NOT NULL,
  `writer` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`book_id`)
);
```



```
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| book_id   | int(11)     | NO   | PRI | NULL    | auto_increment |
| book_name | varchar(40) | NO   |     | NULL    |                |
| writer    | varchar(30) | NO   |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
```

INSERT INTO `Books` VALUES

('1000', 'To Kill a Mockingbird', 'Harper Lee'),

('1001', '1984', 'George Orwell'),

('1002', 'Harry Potter and the Philosopher\'s Stone', 'J.K. Rowling'),

('1003', 'the Lord of the Rings', 'J.R.R'),

('1004', 'the Great Gatsby', 'F. Scott Fitzgerald'),

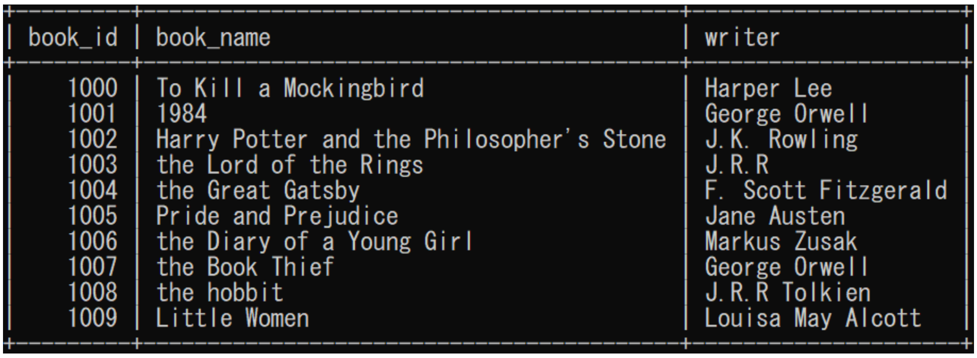
('1005', 'Pride and Prejudice', 'Jane Austen'),

('1006', 'the Diary of a Young Girl', 'Markus Zusak'),

('1007', 'the Book Thief', 'George Orwell'),

('1008', 'the hobbit', 'J.R.R Tolkien'),

('1009', 'Little Women', 'Louisa May Alcott');



```
+---------+------------------------------------------+---------------------+
| book_id | book_name                                | writer              |
+---------+------------------------------------------+---------------------+
|    1000 | To Kill a Mockingbird                    | Harper Lee          |
|    1001 | 1984                                     | George Orwell       |
|    1002 | Harry Potter and the Philosopher's Stone | J.K. Rowling        |
|    1003 | the Lord of the Rings                    | J.R.R               |
|    1004 | the Great Gatsby                         | F. Scott Fitzgerald |
|    1005 | Pride and Prejudice                      | Jane Austen         |
|    1006 | the Diary of a Young Girl                | Markus Zusak        |
|    1007 | the Book Thief                           | George Orwell       |
|    1008 | the hobbit                               | J.R.R Tolkien       |
|    1009 | Little Women                             | Louisa May Alcott   |
+---------+------------------------------------------+---------------------+
```

## Q. issued_books Table

CREATE TABLE `issued_books` (

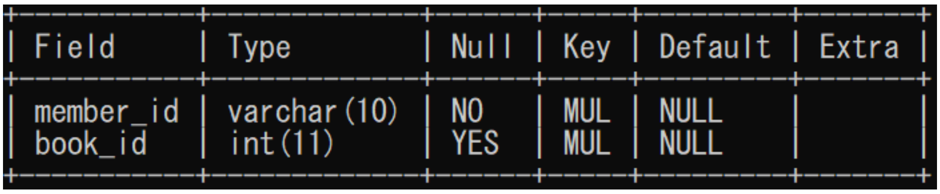
`member_id` VARCHAR(10) NOT NULL,

`book_id` INT ,

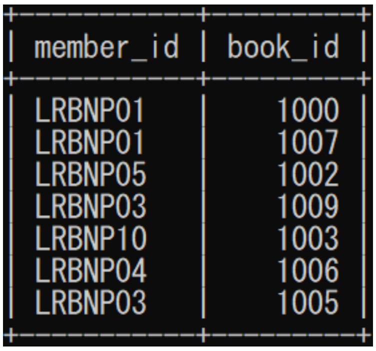
FOREIGN KEY (`member_id`) REFERENCES Members(`member_id`),

FOREIGN KEY (`book_id`) REFERENCES Books(`book_id`)

);



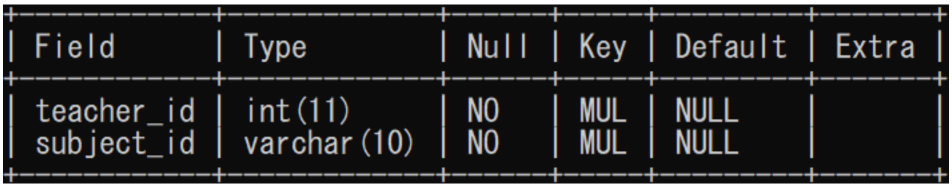
```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| member_id | varchar(10) | NO   | MUL | NULL    |       |
| book_id   | int(11)     | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
INSERT INTO `issued_books` VALUES
('LRBNP01','1000'), ('LRBNP01','1007'), ('LRBNP05','1002'),
('LRBNP03','1009'), ('LRBNP10','1003'), ('LRBNP04','1006'),
('LRBNP03','1005');
```



```
+-----------+---------+
| member_id | book_id |
+-----------+---------+
| LRBNP01   |    1000 |
| LRBNP01   |    1007 |
| LRBNP05   |    1002 |
| LRBNP03   |    1009 |
| LRBNP10   |    1003 |
| LRBNP04   |    1006 |
| LRBNP03   |    1005 |
+-----------+---------+
```

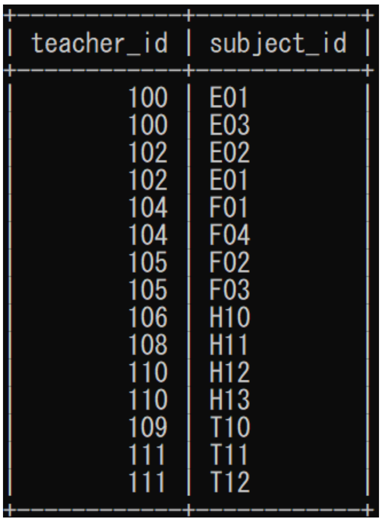
### R. teacher_teaches Table

```
CREATE TABLE `teacher_teaches` (
  `teacher_id` INT NOT NULL,
  `subject_id` VARCHAR(10) NOT NULL,
  FOREIGN KEY (`teacher_id`) REFERENCES Teacher(`teacher_id`),
  FOREIGN KEY (`subject_id`) REFERENCES Subject(`subject_id`)
);
```



```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| teacher_id | int(11)     | NO   | MUL | NULL    |       |
| subject_id | varchar(10) | NO   | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

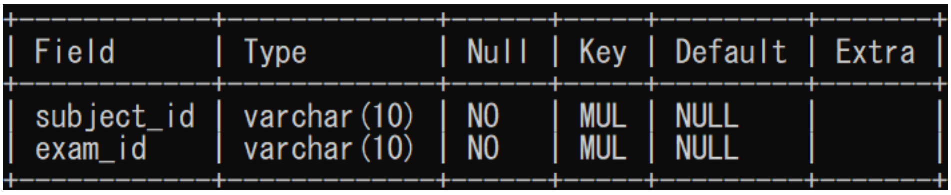
```
INSERT INTO `teacher_teaches` VALUES
('100','E01'), ('100','E03'),
('102','E02'), ('102','E01'),
('104','F01'), ('104','F04'),
('105','F02'), ('105','F03'),
('106','H10'),
('108','H11'),
('110','H12'), ('110','H13'),
('109','T10'),
('111','T11'), ('111','T12');
```



```
+------------+------------+
| teacher_id | subject_id |
+------------+------------+
|        100 | E01        |
|        100 | E03        |
|        102 | E02        |
|        102 | E01        |
|        104 | F01        |
|        104 | F04        |
|        105 | F02        |
|        105 | F03        |
|        106 | H10        |
|        108 | H11        |
|        110 | H12        |
|        110 | H13        |
|        109 | T10        |
|        111 | T11        |
|        111 | T12        |
+------------+------------+
```

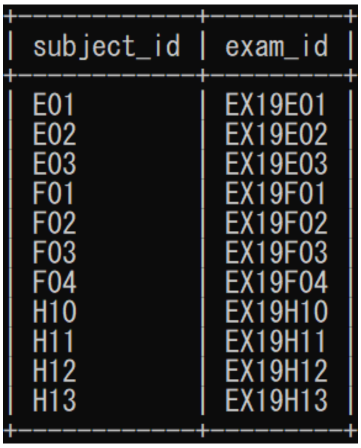
## S. exam_subject Table

```
CREATE TABLE `exam_subject` (
  `subject_id` VARCHAR(10) NOT NULL,
  `exam_id` VARCHAR(10) NOT NULL,
  FOREIGN KEY (`subject_id`) REFERENCES Subject(`subject_id`),
  FOREIGN KEY (`exam_id`) REFERENCES Eximination(`exam_id`)
);
```



```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| subject_id | varchar(10) | NO   | MUL | NULL    |       |
| exam_id    | varchar(10) | NO   | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

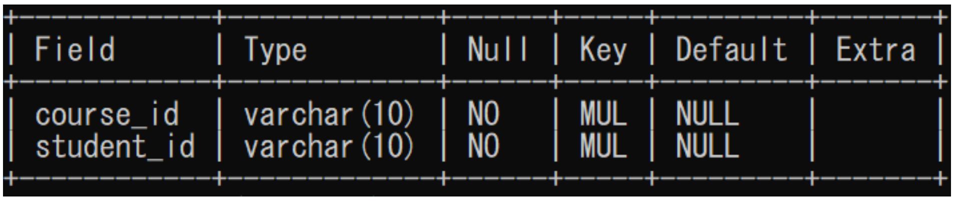
```
INSERT INTO `exam_subject` VALUES
('E01','EX19E01'), ('E02','EX19E02'), ('E03','EX19E03'),
('F01','EX19F01'), ('F02','EX19F02'), ('F03','EX19F03'), ('F04','EX19F04'),
('H10','EX19H10'), ('H11','EX19H11'), ('H12','EX19H12'), ('H13','EX19H13');
```



```
+------------+---------+
| subject_id | exam_id |
+------------+---------+
| E01        | EX19E01 |
| E02        | EX19E02 |
| E03        | EX19E03 |
| F01        | EX19F01 |
| F02        | EX19F02 |
| F03        | EX19F03 |
| F04        | EX19F04 |
| H10        | EX19H10 |
| H11        | EX19H11 |
| H12        | EX19H12 |
| H13        | EX19H13 |
+------------+---------+
```

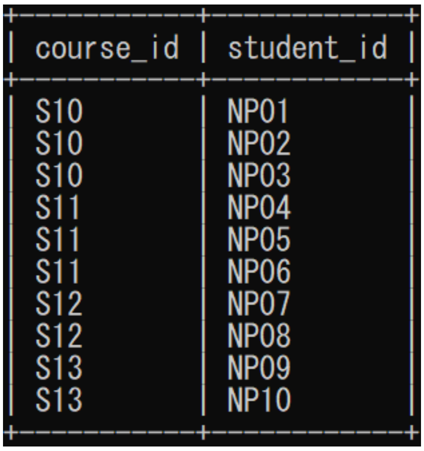
## T. student_course Table

```
CREATE TABLE `student_course` (
  `course_id` VARCHAR(10) NOT NULL,
  `student_id` VARCHAR(10) NOT NULL,
  FOREIGN KEY (`course_id`) REFERENCES Courses(`course_id`),
  FOREIGN KEY (`student_id`) REFERENCES Student(`student_id`)
);
```



```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| course_id  | varchar(10) | NO   | MUL | NULL    |       |
| student_id | varchar(10) | NO   | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

```
INSERT INTO `student_course` VALUES
('S10','NP01'), ('S10','NP02'), ('S10','NP03'),
('S11','NP04'), ('S11','NP05'), ('S11','NP06'),
('S12','NP07'), ('S12','NP08'), ('S13','NP09'), ('S13','NP10');
```



```
+-----------+------------+
| course_id | student_id |
+-----------+------------+
| S10       | NP01       |
| S10       | NP02       |
| S10       | NP03       |
| S11       | NP04       |
| S11       | NP05       |
| S11       | NP06       |
| S12       | NP07       |
| S12       | NP08       |
| S13       | NP09       |
| S13       | NP10       |
+-----------+------------+
```

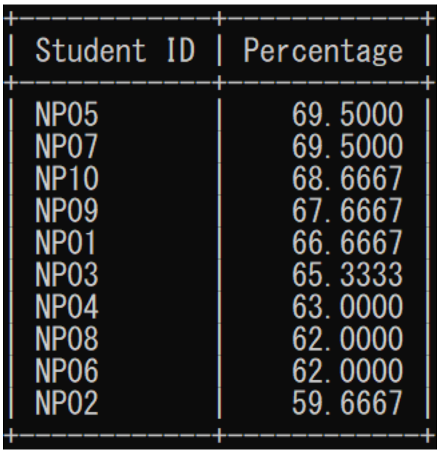
## V. Select Statements using Different Functions

1. Write a query to display total percentage of all student along with their student id.

a. Solution

```
SELECT
`student_id` AS `Student ID`,
SUM(`sub_marks`)/(COUNT(`subject_id`)*100)*100 AS `Percentage`
FROM `Marks`
GROUP BY `student_id`
ORDER BY `Percentage`
DESC;
```

b. Result



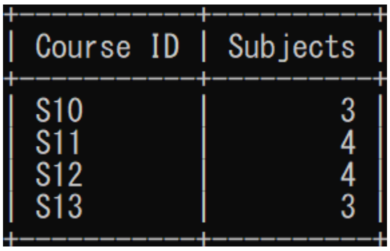
```
+------------+------------+
| Student ID | Percentage |
+------------+------------+
| NP05       |    69.5000 |
| NP07       |    69.5000 |
| NP10       |    68.6667 |
| NP09       |    67.6667 |
| NP01       |    66.6667 |
| NP03       |    65.3333 |
| NP04       |    63.0000 |
| NP08       |    62.0000 |
| NP06       |    62.0000 |
| NP02       |    59.6667 |
+------------+------------+
```

2. Write a query to display total number of subject in each courses

a. Solution

```
SELECT
`course_id` AS `Course ID`,
COUNT(*) AS `Subjects`
FROM `Subject`
Group BY `course_id`;
```

b. Result



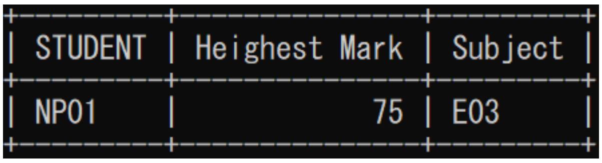
```
+-----------+----------+
| Course ID | Subjects |
+-----------+----------+
| S10       |        3 |
| S11       |        4 |
| S12       |        4 |
| S13       |        3 |
+-----------+----------+
```

3. Write a query to display highest marks obtained in subject E03 along with student id.

a. Solution

```
SELECT
`student_id` AS `STUDENT`,
`sub_marks` AS  `Heighest Mark`,
`subject_id` AS `Subject`
From `marks`
WHERE
`sub_marks` = (SELECT
MAX(`sub_marks`)
FROM `Marks`
WHERE `subject_id`= 'E03') AND `subject_id`= 'E03';
```

b. Result



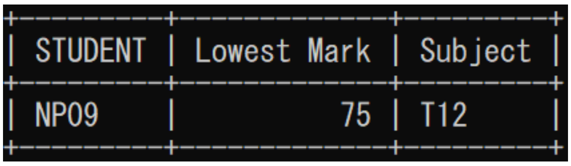
```
+---------+---------------+---------+
| STUDENT | Heighest Mark | Subject |
+---------+---------------+---------+
| NP01    |            75 | E03     |
+---------+---------------+---------+
```

4. Write a query to display lowest marks obtained in subject T12 along with student id.

a. Solution

```
SELECT
`student_id` AS `STUDENT`,
`sub_marks` AS  `Heighest Mark`,
`subject_id` AS `Subject`
From `marks`
WHERE
`sub_marks` = (SELECT
MIN(`sub_marks`)
FROM `Marks`
WHERE `subject_id`= 'T12') AND `subject_id`= 'T12';
```

b. Result



```
+---------+-------------+---------+
| STUDENT | Lowest Mark | Subject |
+---------+-------------+---------+
| NP09    |          75 | T12     |
+---------+-------------+---------+
```

5. Write a query to display Date along with exam id which was held between '2019-05-9' AND '2019-05-10'.

a. Solution

SELECT

DATE(`exam_date`) AS `EXAM DATE`,

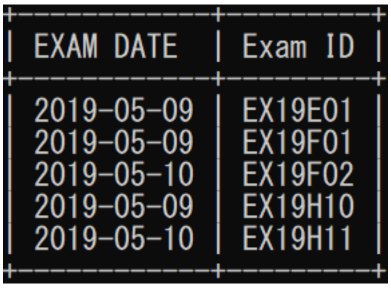
`exam_id` AS `Exam ID`

FROM `eximination`

WHERE `exam_date`

BETWEEN '2019-05-9' AND '2019-05-10';

b. Result



```
+------------+---------+
| EXAM DATE  | Exam ID |
+------------+---------+
| 2019-05-09 | EX19E01 |
| 2019-05-09 | EX19F01 |
| 2019-05-10 | EX19F02 |
| 2019-05-09 | EX19H10 |
| 2019-05-10 | EX19H11 |
+------------+---------+
```

6. Write a query to display dates school had an exam to take in 2019.

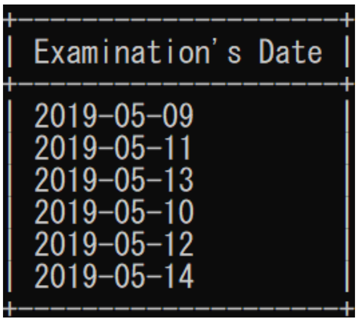
a. Solution

SELECT DISTINCT

`exam_date` AS `Examination's Date`

from `eximination` WHERE `exam_date` LIKE '2019%';

b. Result



```
+--------------------+
| Examination's Date |
+--------------------+
| 2019-05-09         |
| 2019-05-11         |
| 2019-05-13         |
| 2019-05-10         |
| 2019-05-12         |
| 2019-05-14         |
+--------------------+
```

7. Write a query to display the student id along with present(%) whose present percentage is greater than 80.

a. Solution

SELECT `student_id` AS `Student ID`,

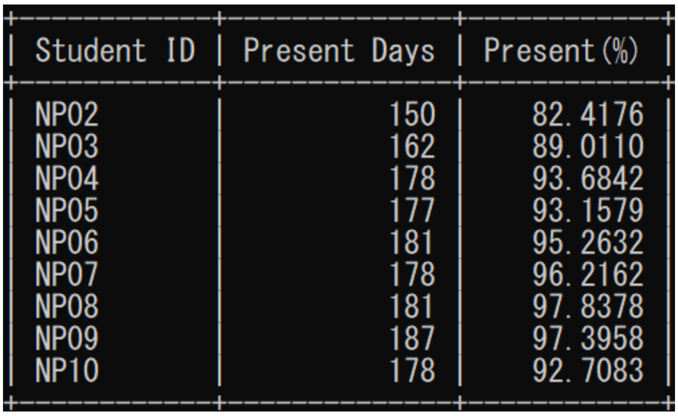
`present_days` AS `Present Days`,

(`present_days`/`total_class`)*100 AS `Present(%)`

FROM `attendance`

WHERE (`present_days`/`total_class`)*100 > 80;

b. Result



```
+------------+--------------+-------------+
| Student ID | Present Days | Present(%)  |
+------------+--------------+-------------+
| NP02       |          150 |     82.4176 |
| NP03       |          162 |     89.0110 |
| NP04       |          178 |     93.6842 |
| NP05       |          177 |     93.1579 |
| NP06       |          181 |     95.2632 |
| NP07       |          178 |     96.2162 |
| NP08       |          181 |     97.8378 |
| NP09       |          187 |     97.3958 |
| NP10       |          178 |     92.7083 |
+------------+--------------+-------------+
```

8. Write a query to display the student is either good, average or poor in subject F04 and E01 respectively.

a. Solution

SELECT

`Student_id` AS `Student ID`,

`subject_id` AS `Subject ID`,

`sub_marks` AS `Subject Marks`,

CASE

WHEN `sub_marks` >= 70 THEN 'Good Student'

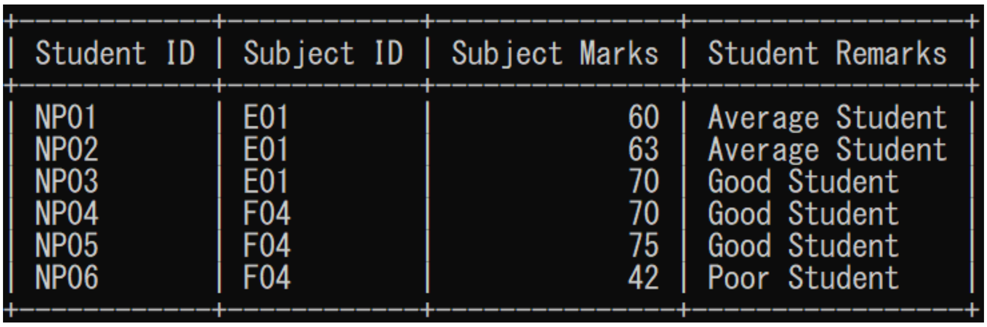
WHEN `sub_marks` >= 60 THEN 'Average Student'

ELSE 'Poor Student'

END AS `Student Remarks`

FROM `Marks` WHERE `subject_id` IN ('F04','E01');

b. Result



```
+------------+------------+---------------+-----------------+
| Student ID | Subject ID | Subject Marks | Student Remarks |
+------------+------------+---------------+-----------------+
| NP01       | E01        |            60 | Average Student |
| NP02       | E01        |            63 | Average Student |
| NP03       | E01        |            70 | Good Student    |
| NP04       | F04        |            70 | Good Student    |
| NP05       | F04        |            75 | Good Student    |
| NP06       | F04        |            42 | Poor Student    |
+------------+------------+---------------+-----------------+
```

9. Write a query to display the total course marks of each student.

a. Solution

SELECT

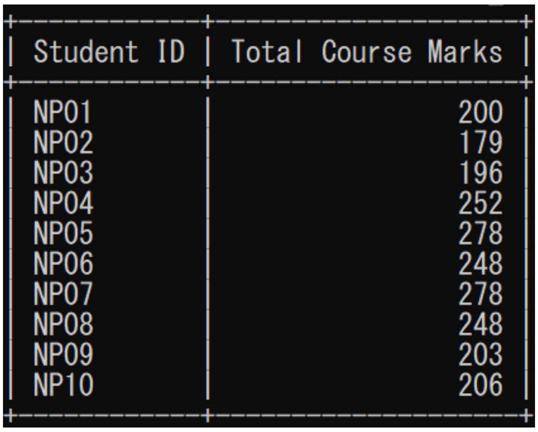
`student_id` AS `Student ID`,

SUM(`sub_marks`) AS `Total Course Marks`

FROM

`Marks` GROUP BY `student_id`;

b. Result



```
+------------+--------------------+
| Student ID | Total Course Marks |
+------------+--------------------+
| NP01       |                200 |
| NP02       |                179 |
| NP03       |                196 |
| NP04       |                252 |
| NP05       |                278 |
| NP06       |                248 |
| NP07       |                278 |
| NP08       |                248 |
| NP09       |                203 |
| NP10       |                206 |
+------------+--------------------+
```

10. Write a query to display teacher name in uppercase letter and in ascending order

a. Solution

SELECT

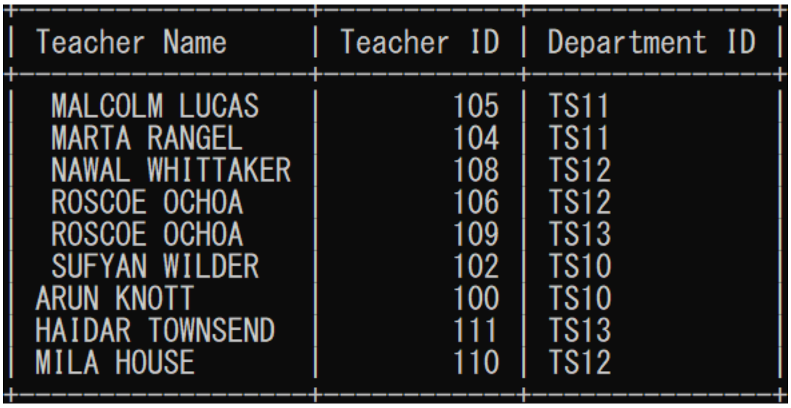
UCASE(`teacher_name`) AS `Teacher Name`,

`teacher_id` AS `Teacher ID`,

`department_id` AS `Department ID`

FROM `Teacher` ORDER BY `teacher_name` ASC;

b. Result



```
+-------------------+------------+---------------+
| Teacher Name      | Teacher ID | Department ID |
+-------------------+------------+---------------+
|  MALCOLM LUCAS    |        105 | TS11          |
|  MARTA RANGEL     |        104 | TS11          |
|  NAWAL WHITTAKER  |        108 | TS12          |
|  ROSCOE OCHOA     |        106 | TS12          |
|  ROSCOE OCHOA     |        109 | TS13          |
|  SUFYAN WILDER    |        102 | TS10          |
| ARUN KNOTT        |        100 | TS10          |
| HAIDAR TOWNSEND   |        111 | TS13          |
| MILA HOUSE        |        110 | TS12          |
+-------------------+------------+---------------+
```

11. Write a query to trim and display teacher name in lowercase letter and in ascending order

a. Solution

SELECT

LCASE(`Teacher Name`),

`Teacher ID`,

`Department ID`

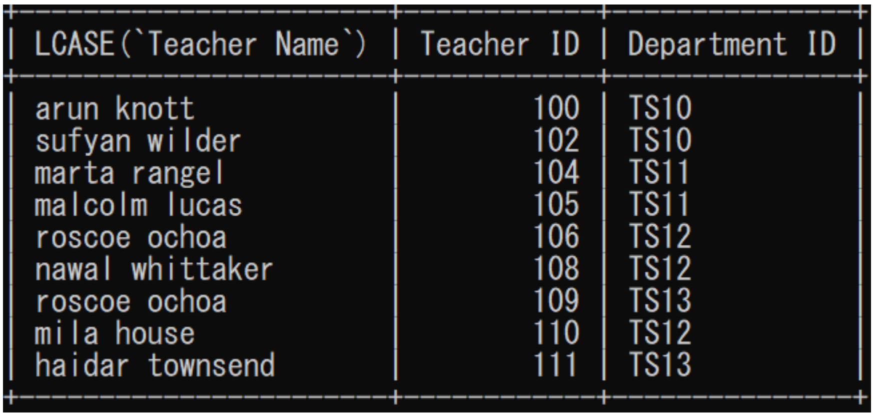
FROM (SELECT TRIM(`teacher_name`) AS `Teacher Name`,

`teacher_id` AS `Teacher ID`,

`department_id` AS `Department ID`

FROM `Teacher` ORDER BY `teacher_name` ASC) AS T;

b. Result



```
+------------------------+-------------+---------------+
| LCASE(`Teacher Name`)  | Teacher ID  | Department ID |
+------------------------+-------------+---------------+
| arun knott             |         100 | TS10          |
| sufyan wilder          |         102 | TS10          |
| marta rangel           |         104 | TS11          |
| malcolm lucas          |         105 | TS11          |
| roscoe ochoa           |         106 | TS12          |
| nawal whittaker        |         108 | TS12          |
| roscoe ochoa           |         109 | TS13          |
| mila house             |         110 | TS12          |
| haidar townsend        |         111 | TS13          |
+------------------------+-------------+---------------+
```

12. Write a query to replace first two letter of student id with ‘PP’ as new student ID

a. Solution

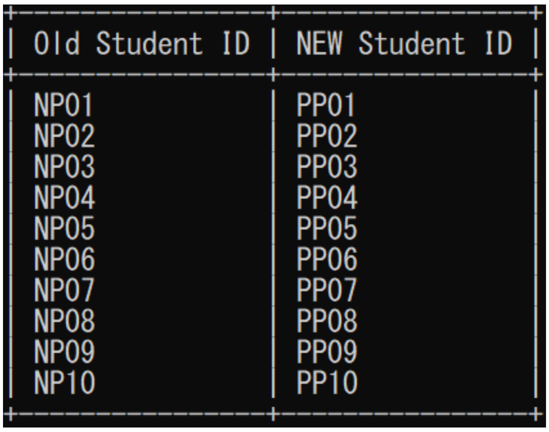
SELECT

`Student_id` AS `Old Student ID`,

REPLACE(`student_id`, 'NP', 'PP') AS `NEW Student ID`

FROM `Student`;

b. Result



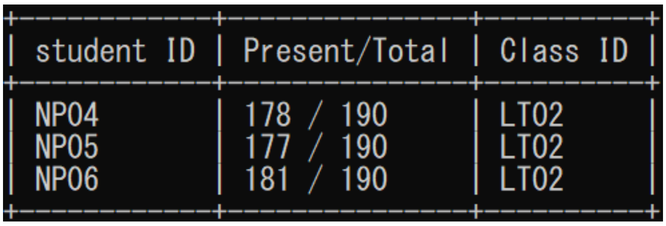
```
+----------------+----------------+
| Old Student ID | NEW Student ID |
+----------------+----------------+
| NP01           | PP01           |
| NP02           | PP02           |
| NP03           | PP03           |
| NP04           | PP04           |
| NP05           | PP05           |
| NP06           | PP06           |
| NP07           | PP07           |
| NP08           | PP08           |
| NP09           | PP09           |
| NP10           | PP10           |
+----------------+----------------+
```

13. Write a query to display present days over total date from table attendance of class LT02

a. Solution

```
SELECT
Student.student_id AS `student ID`,
CONCAT(`present_days`,' \/ ',`total_class`) AS `Present/Total`,
`class_id` AS `Class ID`
FROM
`student`, `attendance`
WHERE
student.student_id = attendance.student_id
AND Student.class_id = 'LT02';
```

b. Result



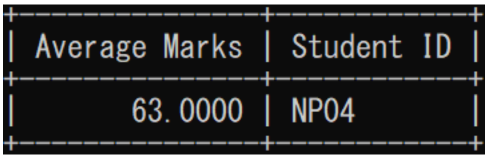
```
+------------+----------------+----------+
| student ID | Present/Total  | Class ID |
+------------+----------------+----------+
| NP04       | 178 / 190      | LT02     |
| NP05       | 177 / 190      | LT02     |
| NP06       | 181 / 190      | LT02     |
+------------+----------------+----------+
```

14. Write a query to display the average marks of student having student id NP04

a. Solution

```
SELECT
AVG(`sub_marks`) AS `Average Marks`,
`student_id` AS `Student ID`
FROM `marks`
WHERE `student_id`='NP04';
```

b. Result



```
+---------------+------------+
| Average Marks | Student ID |
+---------------+------------+
|       63.0000 | NP04       |
+---------------+------------+
```

15. Write a query to count total number of books in library.

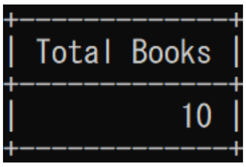
a. Solution

SELECT

COUNT(`book_id`) AS `Total Books`

FROM `books`;

b. Result



```
+-------------+
| Total Books |
+-------------+
|          10 |
+-------------+
```

16. Write a query to extract 3 letters from left of student name.

a. Solution

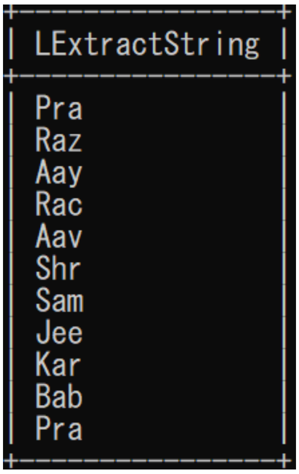
SELECT

LEFT(`student_name`, 3)

AS `LExtractString`

FROM `student`;

b. Result



```
+----------------+
| LExtractString |
+----------------+
| Pra            |
| Raz            |
| Aay            |
| Rac            |
| Aav            |
| Shr            |
| Sam            |
| Jee            |
| Kar            |
| Bab            |
| Pra            |
+----------------+
```

17. Write a query to extract 3 letters from right of student name.

a. Solution

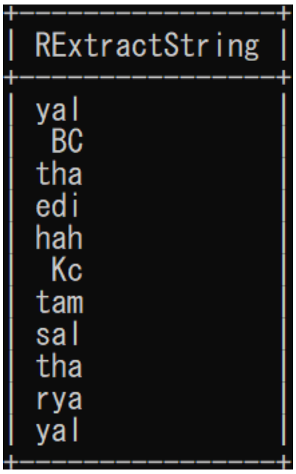
SELECT

RIGHT(`student_name`, 3)

AS `RExtractString`

FROM `student`;

b. Result



```
+----------------+
| RExtractString |
+----------------+
| yal            |
|  BC            |
| tha            |
| edi            |
| hah            |
|  Kc            |
| tam            |
| sal            |
| tha            |
| rya            |
| yal            |
+----------------+
```

18. Write a query to extract 3 letters from middle of student name along with their student id.

a. Solution

SELECT

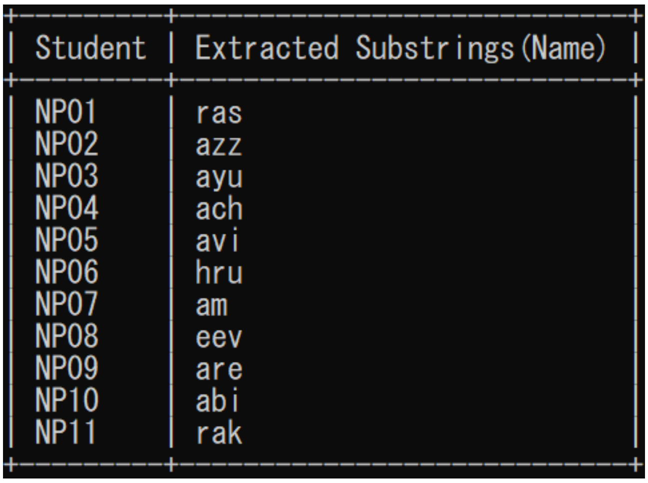
`student_id` AS `Student`,

MID(Student_name, 2, 3)

AS `Extracted Substrings(Name)`

FROM Student;

b. Result



```
+---------+----------------------------+
| Student | Extracted Substrings(Name) |
+---------+----------------------------+
| NP01    | ras                        |
| NP02    | azz                        |
| NP03    | ayu                        |
| NP04    | ach                        |
| NP05    | avi                        |
| NP06    | hru                        |
| NP07    | am                         |
| NP08    | eev                        |
| NP09    | are                        |
| NP10    | abi                        |
| NP11    | rak                        |
+---------+----------------------------+
```

19. Write a query to display name of book in top of books table when ordered by book id.

a. Solution

Select

`book_name` AS `First Book in Table`

FROM

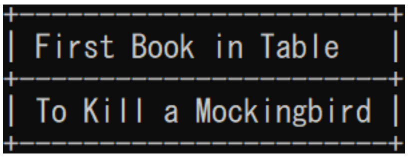
`Books`

ORDER BY `book_id`

ASC

LIMIT 1;

b. Result



```
+-----------------------+
| First Book in Table   |
+-----------------------+
| To Kill a Mockingbird |
+-----------------------+
```

20. Write a query to display name of book in bottom of books table when ordered by book id.

a. Solution

Select

`book_name` AS `Last Book in Table`

FROM

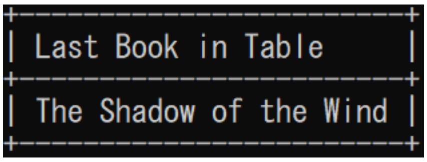
`Books`

ORDER BY `book_id`

DESC

LIMIT 1;

b. Result



```
+------------------------+
| Last Book in Table     |
+------------------------+
| The Shadow of the Wind |
+------------------------+
```

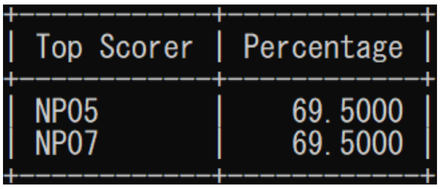
## VI. Select Statements using Sub Query

1. Write a query to display total percentage and find the student with highest score among them.

a. Solution

```
SELECT
`student_id` AS `Top Scorer`,
SUM(`sub_marks`)/(COUNT(`subject_id`)*100)*100 AS `Percentage`
FROM `Marks`
GROUP BY `student_id`
HAVING (SUM(`sub_marks`)/(COUNT(`subject_id`)*100)*100) =
(
SELECT MAX(`Percentage`)
FROM (
SELECT
`student_id` AS `Student ID`,
SUM(`sub_marks`)/(COUNT(`subject_id`)*100)*100 AS `Percentage`
FROM `Marks`
GROUP BY `student_id`
ORDER BY `Percentage` DESC) AS `MaxGrade`
);
```

b. Result



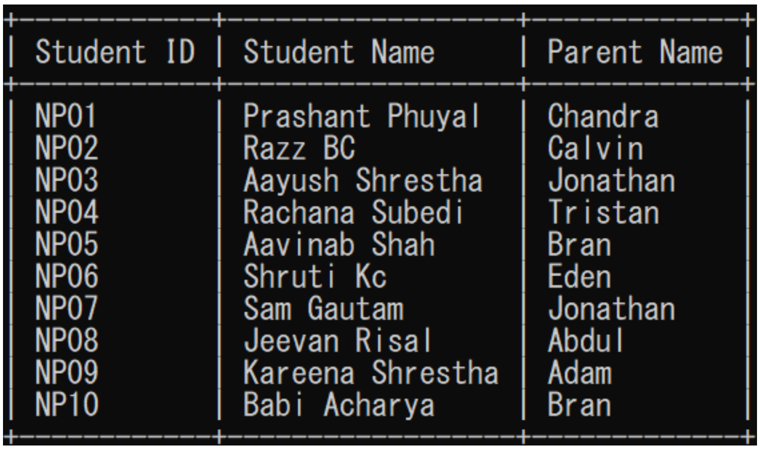
```
+------------+------------+
| Top Scorer | Percentage |
+------------+------------+
| NP05       |    69.5000 |
| NP07       |    69.5000 |
+------------+------------+
```

2. Write a query to display Student ID, Student name, Parent Name after joining two tables parent and student using sub query.

a. Solution

```
SELECT
`student_id` AS `Student ID`,
`student_name` AS `Student Name`,
`parent_name` AS `Parent Name`
FROM (
SELECT
Student.student_id,
Student.student_name,
Parent.parent_name,
Parent.phone_number,
Student.class_id
FROM `Student`
INNER JOIN `Parent`
ON Student.student_id = Parent.student_id
) AS JoinedTable;
```

b. Result



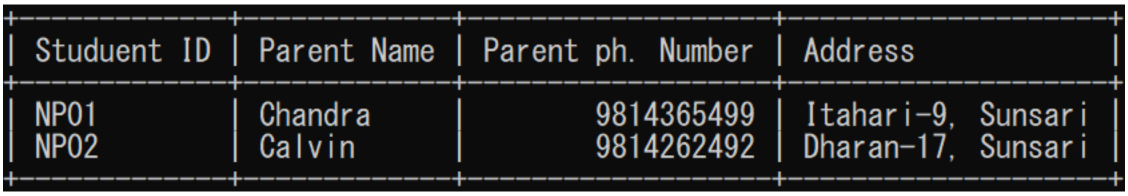
```
+------------+------------------+-------------+
| Student ID | Student Name     | Parent Name |
+------------+------------------+-------------+
| NP01       | Prashant Phuyal  | Chandra     |
| NP02       | Razz BC          | Calvin      |
| NP03       | Aayush Shrestha  | Jonathan    |
| NP04       | Rachana Subedi   | Tristan     |
| NP05       | Aavinab Shah     | Bran        |
| NP06       | Shruti Kc        | Eden        |
| NP07       | Sam Gautam       | Jonathan    |
| NP08       | Jeevan Risal     | Abdul       |
| NP09       | Kareena Shrestha | Adam        |
| NP10       | Babi Acharya     | Bran        |
+------------+------------------+-------------+
```

3. Write a query to display student ID, Parent Name, Phone Number, parent address to parent name starting with C using sub query.

a. Solution

```
SELECT
`student_id` AS `Studuent ID`,
`parent_name` AS `Parent Name`,
`phone_number` AS `Parent ph. Number`,
`parent_address` AS `Address`
FROM `Parent`
WHERE `parent_name`
IN (
SELECT `parent_name`
FROM `parent`
WHERE `parent_name` LIKE 'C%'
);
```

b. Result



```
+-------------+-------------+-------------------+---------------------+
| Studuent ID | Parent Name | Parent ph. Number | Address             |
+-------------+-------------+-------------------+---------------------+
| NP01        | Chandra     |        9814365499 | Itahari-9, Sunsari  |
| NP02        | Calvin      |        9814262492 | Dharan-17, Sunsari  |
+-------------+-------------+-------------------+---------------------+
```

## VII. Select Statements using Count and Group Functions

1. Write a query to display total percentage of all students.

a. Solution

SELECT

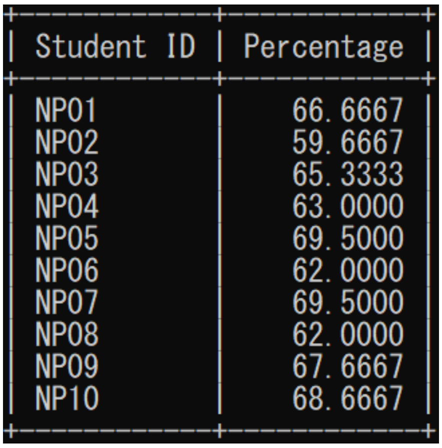
`student_id` AS `Student ID`,

SUM(`sub_marks`)/(COUNT(`subject_id`)*100)*100 AS `Percentage`

FROM `Marks`

GROUP BY `student_id`;

b. Result



```
+------------+------------+
| Student ID | Percentage |
+------------+------------+
| NP01       |    66.6667 |
| NP02       |    59.6667 |
| NP03       |    65.3333 |
| NP04       |    63.0000 |
| NP05       |    69.5000 |
| NP06       |    62.0000 |
| NP07       |    69.5000 |
| NP08       |    62.0000 |
| NP09       |    67.6667 |
| NP10       |    68.6667 |
+------------+------------+
```

2. Write a query to display all the departments along with the total number of teachers in each department.

a. Solution

SELECT

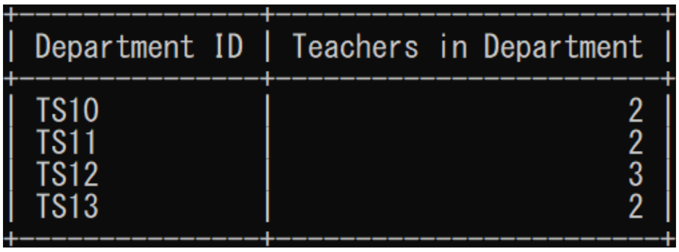
`department_id` AS `Department ID`,

COUNT(*) AS `Teachers in Department`

FROM `Teacher`

Group BY `department_id`;

b. Result



```
+---------------+------------------------+
| Department ID | Teachers in Department |
+---------------+------------------------+
| TS10          |                      2 |
| TS11          |                      2 |
| TS12          |                      3 |
| TS13          |                      2 |
+---------------+------------------------+
```

## VIII. Select Statements using Different Joins

1. Write a query to join table's student, Courses and class with attributes Student ID, Course ID, and Class ID Using INNER JOIN.

a. Solution

SELECT

S.student_id `Student ID`,

C.course_id `Course ID`,

Cl.class_id `Class ID`

FROM

Student AS S

INNER JOIN

student_course AS sc

ON S.student_id = Sc.student_id

INNER JOIN

Courses AS C

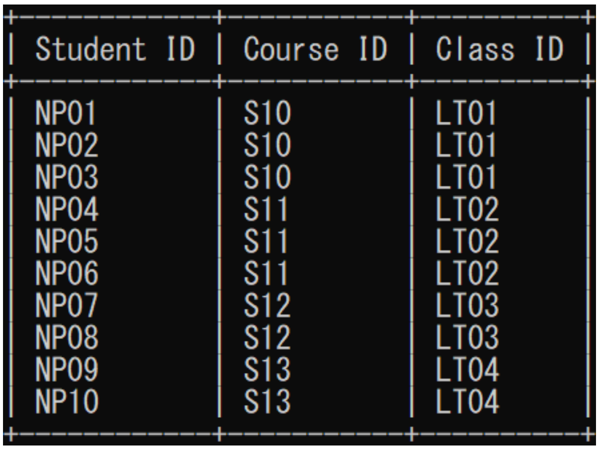
ON C.course_id = Sc.course_id

INNER JOIN

Class AS Cl

ON Cl.class_id = S.class_id;

b. Result



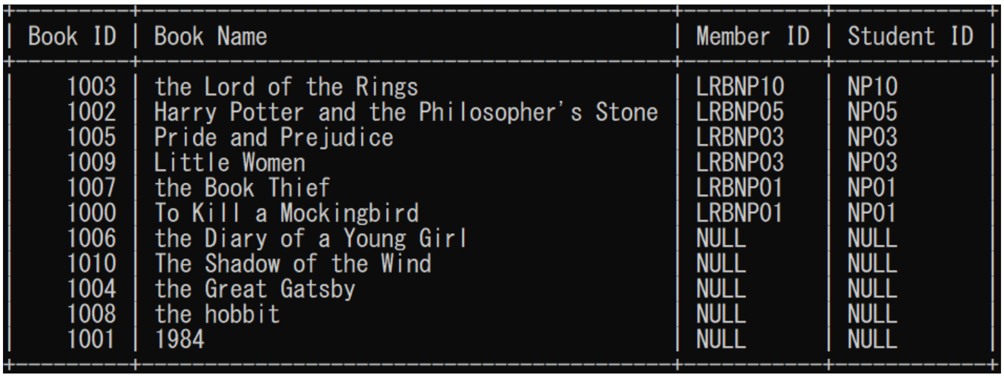
```
+------------+-----------+----------+
| Student ID | Course ID | Class ID |
+------------+-----------+----------+
| NP01       | S10       | LT01     |
| NP02       | S10       | LT01     |
| NP03       | S10       | LT01     |
| NP04       | S11       | LT02     |
| NP05       | S11       | LT02     |
| NP06       | S11       | LT02     |
| NP07       | S12       | LT03     |
| NP08       | S12       | LT03     |
| NP09       | S13       | LT04     |
| NP10       | S13       | LT04     |
+------------+-----------+----------+
```

2. Write a query to join two tables Books and members with attributes Student ID, Member ID, Books ID and books name using LEFT JOIN.

a. Solution

```
SELECT
B.book_id `Book ID`,
B.book_name `Book Name`,
M.member_id `Member ID`,
M.student_id `Student ID`
FROM
Books AS B
LEFT JOIN
issued_books AS I
ON B.book_id = I.book_id
LEFT JOIN
Members AS M
ON
M.member_id = I.member_id ORDER BY M.member_id DESC;
```

b. Result



```
+---------+-------------------------------------------+-----------+------------+
| Book ID | Book Name                                 | Member ID | Student ID |
+---------+-------------------------------------------+-----------+------------+
|    1003 | the Lord of the Rings                     | LRBNP10   | NP10       |
|    1002 | Harry Potter and the Philosopher's Stone  | LRBNP05   | NP05       |
|    1005 | Pride and Prejudice                       | LRBNP03   | NP03       |
|    1009 | Little Women                              | LRBNP03   | NP03       |
|    1007 | the Book Thief                            | LRBNP01   | NP01       |
|    1000 | To Kill a Mockingbird                     | LRBNP01   | NP01       |
|    1006 | the Diary of a Young Girl                 | NULL      | NULL       |
|    1010 | The Shadow of the Wind                    | NULL      | NULL       |
|    1004 | the Great Gatsby                          | NULL      | NULL       |
|    1008 | the hobbit                                | NULL      | NULL       |
|    1001 | 1984                                      | NULL      | NULL       |
+---------+-------------------------------------------+-----------+------------+
```

3. Write a query to join tables Books and members with attributes Student ID, Member ID, Books ID and books name using RIGHT JOIN.

a. Solution

```
SELECT
B.book_id `Book ID`,
B.book_name `Book Name`,
M.member_id `Member ID`,
M.student_id `Student ID`
FROM
Books AS B
RIGHT JOIN
```

issued_books AS I

ON B.book_id = I.book_id

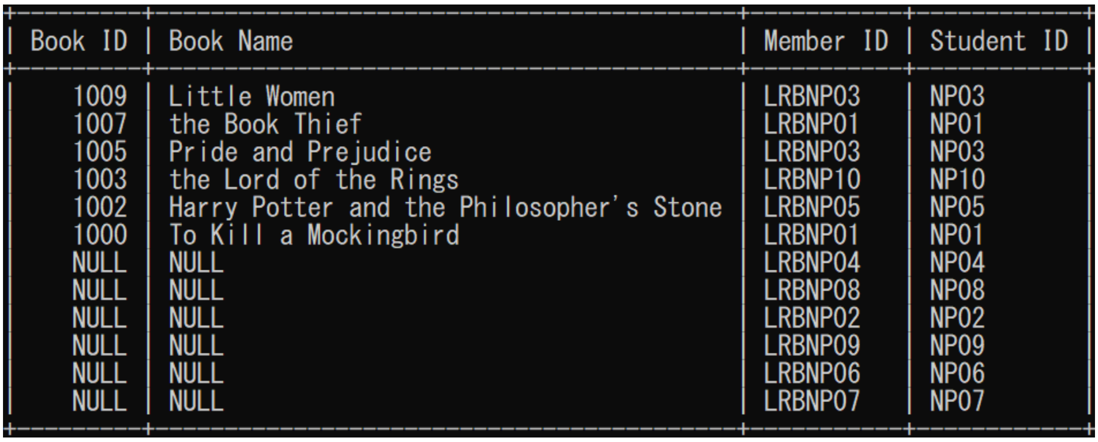
RIGHT JOIN

Members AS M

ON

M.member_id = I.member_id ORDER BY B.book_id DESC;

b. Result



```
+---------+----------------------------------------+-----------+------------+
| Book ID | Book Name                              | Member ID | Student ID |
+---------+----------------------------------------+-----------+------------+
|    1009 | Little Women                           | LRBNP03   | NP03       |
|    1007 | the Book Thief                         | LRBNP01   | NP01       |
|    1005 | Pride and Prejudice                    | LRBNP03   | NP03       |
|    1003 | the Lord of the Rings                  | LRBNP10   | NP10       |
|    1002 | Harry Potter and the Philosopher's Stone | LRBNP05   | NP05       |
|    1000 | To Kill a Mockingbird                  | LRBNP01   | NP01       |
|    NULL | NULL                                   | LRBNP04   | NP04       |
|    NULL | NULL                                   | LRBNP08   | NP08       |
|    NULL | NULL                                   | LRBNP02   | NP02       |
|    NULL | NULL                                   | LRBNP09   | NP09       |
|    NULL | NULL                                   | LRBNP06   | NP06       |
|    NULL | NULL                                   | LRBNP07   | NP07       |
+---------+----------------------------------------+-----------+------------+
```

4. Write a query to join tables Department, Teachers, Subject and Period Hour with attributes Teacher ID, Teacher Name, Department ID, Subject ID and Time period using JOIN.

a. Solution

SELECT

T.teacher_id AS `Teacher ID`,

T.teacher_name `Teacher Name`,

D.department_id `Department ID`,

S.subject_id `Subejct ID`,

PH.time_period `Time`

FROM

Department AS D

JOIN

Teacher AS T

ON D.department_id = T.department_id

JOIN

teacher_teaches AS TT

ON T.teacher_id = TT.teacher_id

JOIN

Subject AS S

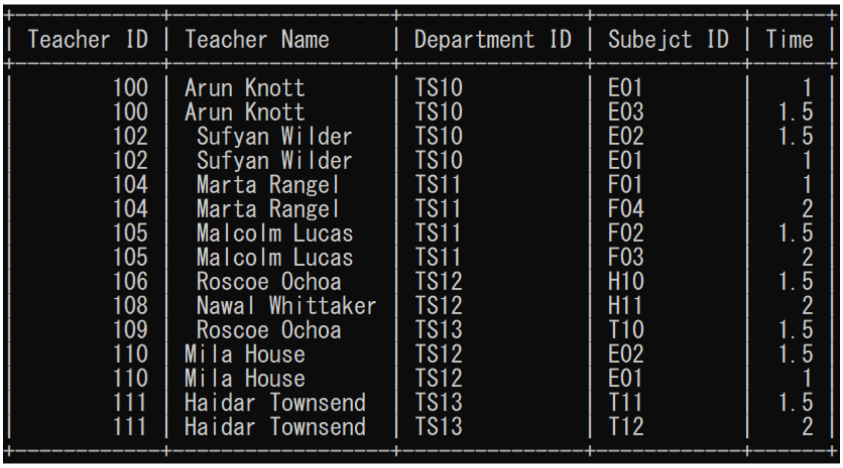
ON S.subject_id = TT.subject_id

JOIN

PeriodHour AS PH

ON S.subject_id = PH.subject_id;

b. Result



```
+------------+--------------------+----------------+------------+------+
| Teacher ID | Teacher Name       | Department ID  | Subejct ID | Time |
+------------+--------------------+----------------+------------+------+
|        100 | Arun Knott         | TS10           | E01        |    1 |
|        100 | Arun Knott         | TS10           | E03        |  1.5 |
|        102 |  Sufyan Wilder     | TS10           | E02        |  1.5 |
|        102 |  Sufyan Wilder     | TS10           | E01        |    1 |
|        104 |  Marta Rangel      | TS11           | F01        |    1 |
|        104 |  Marta Rangel      | TS11           | F04        |    2 |
|        105 |  Malcolm Lucas     | TS11           | F02        |  1.5 |
|        105 |  Malcolm Lucas     | TS11           | F03        |    2 |
|        106 |  Roscoe Ochoa      | TS12           | H10        |  1.5 |
|        108 |  Nawal Whittaker   | TS12           | H11        |    2 |
|        109 |  Roscoe Ochoa      | TS13           | T10        |  1.5 |
|        110 | Mila House         | TS12           | E02        |  1.5 |
|        110 | Mila House         | TS12           | E01        |    1 |
|        111 | Haidar Townsend    | TS13           | T11        |  1.5 |
|        111 | Haidar Townsend    | TS13           | T12        |    2 |
+------------+--------------------+----------------+------------+------+
```

5. Write a query to join tables Student, Eximination, Subject and class with attributes Exam ID, Student ID and Subject ID without using JOIN.

a. Solution

SELECT

E.exam_id `Exam ID`,

S.student_id `Student ID`,

Su.subject_id `Subject ID`

FROM

Student AS S,

Eximination AS E,

Subject AS Su,

Class AS Cl,

exam_subject AS Es

WHERE

S.class_id = Cl.class_id

AND

Cl.class_id = Su.class_id

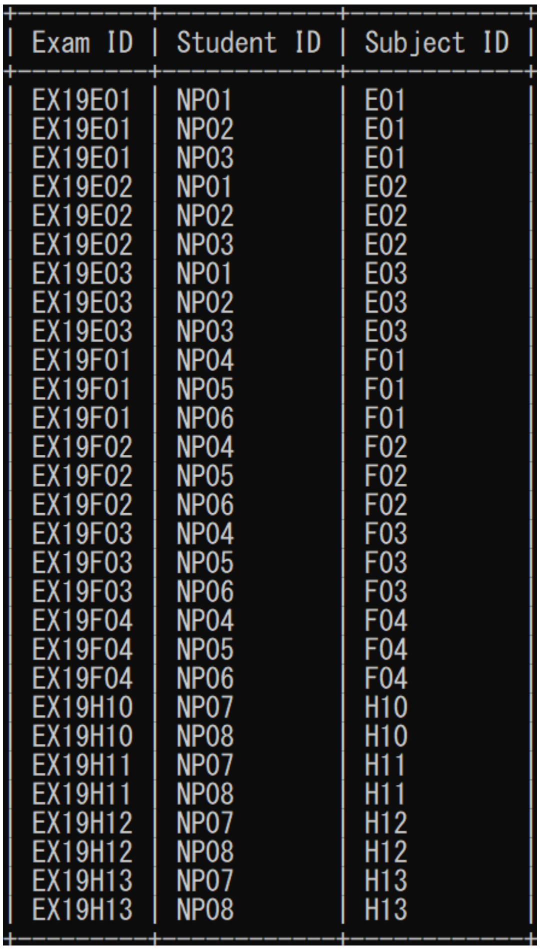
AND

Su.subject_id = Es.subject_id

AND

E.exam_id = Es.exam_id;

b. Result



```
+----------+-------------+------------+
| Exam ID  | Student ID  | Subject ID |
+----------+-------------+------------+
| EX19E01  | NP01        | E01        |
| EX19E01  | NP02        | E01        |
| EX19E01  | NP03        | E01        |
| EX19E02  | NP01        | E02        |
| EX19E02  | NP02        | E02        |
| EX19E02  | NP03        | E02        |
| EX19E03  | NP01        | E03        |
| EX19E03  | NP02        | E03        |
| EX19E03  | NP03        | E03        |
| EX19F01  | NP04        | F01        |
| EX19F01  | NP05        | F01        |
| EX19F01  | NP06        | F01        |
| EX19F02  | NP04        | F02        |
| EX19F02  | NP05        | F02        |
| EX19F02  | NP06        | F02        |
| EX19F03  | NP04        | F03        |
| EX19F03  | NP05        | F03        |
| EX19F03  | NP06        | F03        |
| EX19F04  | NP04        | F04        |
| EX19F04  | NP05        | F04        |
| EX19F04  | NP06        | F04        |
| EX19H10  | NP07        | H10        |
| EX19H10  | NP08        | H10        |
| EX19H11  | NP07        | H11        |
| EX19H11  | NP08        | H11        |
| EX19H12  | NP07        | H12        |
| EX19H12  | NP08        | H12        |
| EX19H13  | NP07        | H13        |
| EX19H13  | NP08        | H13        |
+----------+-------------+------------+
```

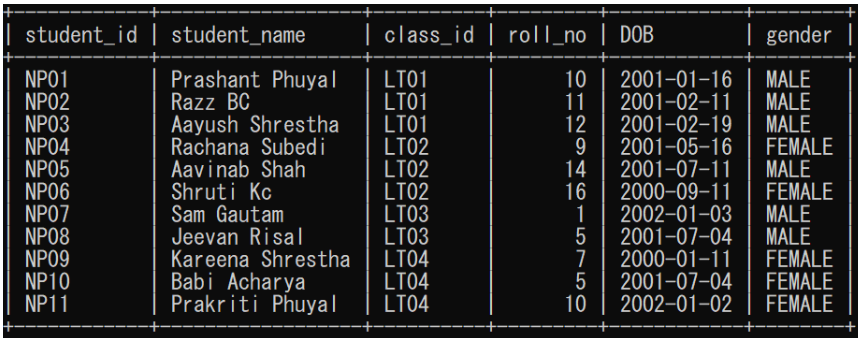
## IX. Insert Statement

1. Write a query to insert student id, student name, class id, roll no, DOB and gender of student table.

a. Solution

```
INSERT INTO `Student` ( `student_id`, `student_name`, `class_id`, `roll_no`, `DOB`,
`gender`)
VALUES (
'NP11','Prakriti Phuyal','LT04','10','2002-01-02','FEMALE'
);
```

b. Result



```
+------------+----------------+----------+---------+------------+--------+
| student_id | student_name   | class_id | roll_no | DOB        | gender |
+------------+----------------+----------+---------+------------+--------+
| NP01       | Prashant Phuyal | LT01    |      10 | 2001-01-16 | MALE   |
| NP02       | Razz BC        | LT01     |      11 | 2001-02-11 | MALE   |
| NP03       | Aayush Shrestha | LT01    |      12 | 2001-02-19 | MALE   |
| NP04       | Rachana Subedi | LT02     |       9 | 2001-05-16 | FEMALE |
| NP05       | Aavinab Shah   | LT02     |      14 | 2001-07-11 | MALE   |
| NP06       | Shruti Kc      | LT02     |      16 | 2000-09-11 | FEMALE |
| NP07       | Sam Gautam     | LT03     |       1 | 2002-01-03 | MALE   |
| NP08       | Jeevan Risal   | LT03     |       5 | 2001-07-04 | MALE   |
| NP09       | Kareena Shrestha | LT04   |       7 | 2000-01-11 | FEMALE |
| NP10       | Babi Acharya   | LT04     |       5 | 2001-07-04 | FEMALE |
| NP11       | Prakriti Phuyal | LT04    |      10 | 2002-01-02 | FEMALE |
+------------+----------------+----------+---------+------------+--------+
```

2. Write a query to insert parent name, student id, phone number, parent address of parent table.

a. Solution

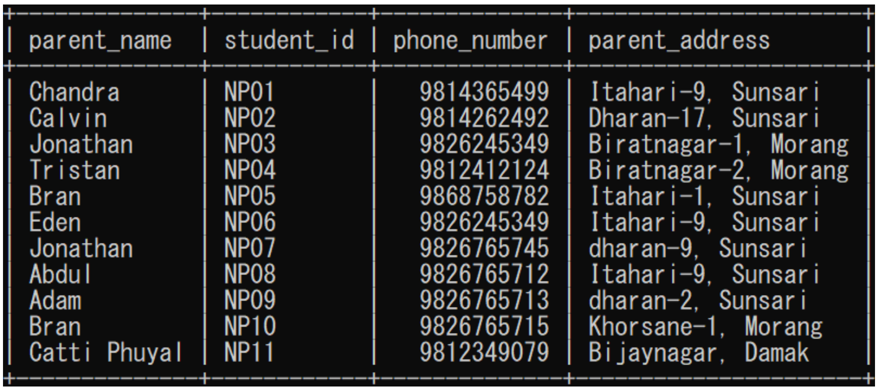
INSERT INTO `Parent` ( `parent_name`, `student_id`, `phone_number`, `parent_address`)

VALUES (

'Catti Phuyal','NP11','9812349079','Bijaynagar, Damak'

);

b. Result



```
+--------------+------------+--------------+---------------------+
| parent_name  | student_id | phone_number | parent_address      |
+--------------+------------+--------------+---------------------+
| Chandra      | NP01       |   9814365499 | Itahari-9, Sunsari  |
| Calvin       | NP02       |   9814262492 | Dharan-17, Sunsari  |
| Jonathan     | NP03       |   9826245349 | Biratnagar-1, Morang |
| Tristan      | NP04       |   9812412124 | Biratnagar-2, Morang |
| Bran         | NP05       |   9868758782 | Itahari-1, Sunsari  |
| Eden         | NP06       |   9826245349 | Itahari-9, Sunsari  |
| Jonathan     | NP07       |   9826765745 | dharan-9, Sunsari   |
| Abdul        | NP08       |   9826765712 | Itahari-9, Sunsari  |
| Adam         | NP09       |   9826765713 | dharan-2, Sunsari   |
| Bran         | NP10       |   9826765715 | Khorsane-1, Morang  |
| Catti Phuyal | NP11       |   9812349079 | Bijaynagar, Damak   |
+--------------+------------+--------------+---------------------+
```

3. Write a query to add student NP11 in enrolling course.

a. Solution

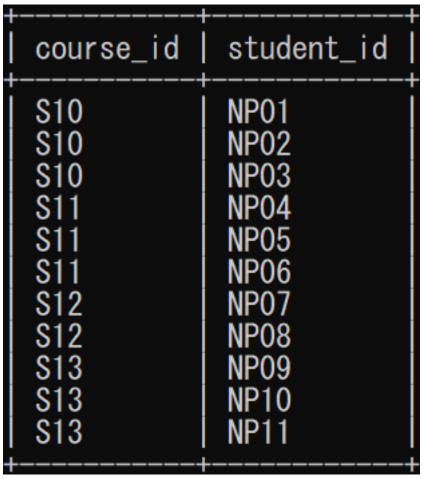
INSERT INTO `student_course` (`course_id`,`student_id`)

VALUES (

'S13', 'NP11'

);

b. Result



```
+-----------+------------+
| course_id | student_id |
+-----------+------------+
| S10       | NP01       |
| S10       | NP02       |
| S10       | NP03       |
| S11       | NP04       |
| S11       | NP05       |
| S11       | NP06       |
| S12       | NP07       |
| S12       | NP08       |
| S13       | NP09       |
| S13       | NP10       |
| S13       | NP11       |
+-----------+------------+
```

4. Write a query to add student NP11 in members table as a member of library.

a. Solution

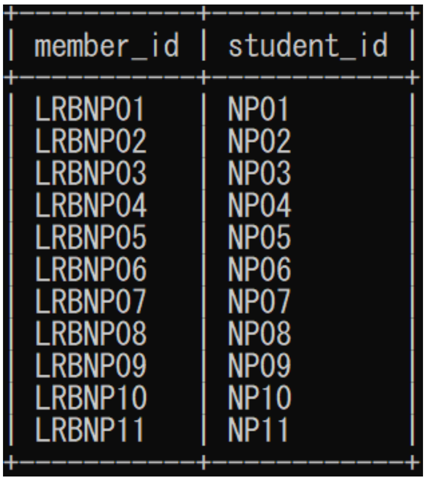
INSERT INTO `Members` (`member_id`,`student_id`)

VALUES (

'LRBNP11', 'NP11'

);

b. Result



```
+-----------+------------+
| member_id | student_id |
+-----------+------------+
| LRBNP01   | NP01       |
| LRBNP02   | NP02       |
| LRBNP03   | NP03       |
| LRBNP04   | NP04       |
| LRBNP05   | NP05       |
| LRBNP06   | NP06       |
| LRBNP07   | NP07       |
| LRBNP08   | NP08       |
| LRBNP09   | NP09       |
| LRBNP10   | NP10       |
| LRBNP11   | NP11       |
+-----------+------------+
```

5. Write a query to insert books in parent table.

a. Solution

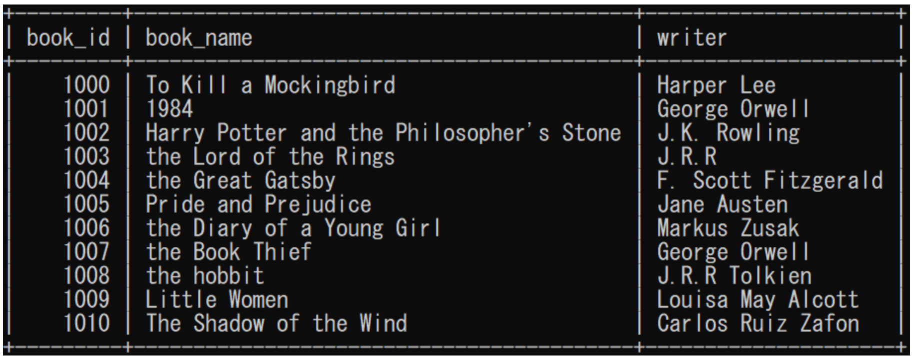
INSERT INTO `Books` (`book_name`,`writer`)

VALUES (

'The Shadow of the Wind', 'Carlos Ruiz Zafon'

);

b. Result



```
+---------+----------------------------------------------+---------------------+
| book_id | book_name                                    | writer              |
+---------+----------------------------------------------+---------------------+
|    1000 | To Kill a Mockingbird                        | Harper Lee          |
|    1001 | 1984                                         | George Orwell       |
|    1002 | Harry Potter and the Philosopher's Stone     | J.K. Rowling        |
|    1003 | the Lord of the Rings                        | J.R.R               |
|    1004 | the Great Gatsby                             | F. Scott Fitzgerald |
|    1005 | Pride and Prejudice                          | Jane Austen         |
|    1006 | the Diary of a Young Girl                    | Markus Zusak        |
|    1007 | the Book Thief                               | George Orwell       |
|    1008 | the hobbit                                   | J.R.R Tolkien       |
|    1009 | Little Women                                 | Louisa May Alcott   |
|    1010 | The Shadow of the Wind                       | Carlos Ruiz Zafon   |
+---------+----------------------------------------------+---------------------+
```

## X. Update Statement

1. Write a query to update student name, class id, gender from student table of student NP07.

a. Solution

UPDATE `Student`

SET

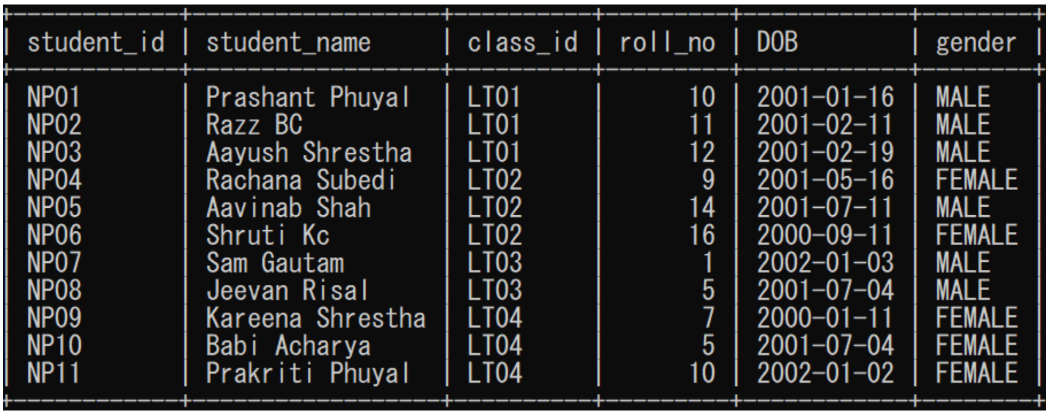
`student_name` = 'Samyog Gotam',

`class_id` = 'LT04',

`gender` = 'FEMALE'

WHERE `student_id` = 'NP07';

b. Result



```
+------------+------------------+----------+---------+------------+--------+
| student_id | student_name     | class_id | roll_no | DOB        | gender |
+------------+------------------+----------+---------+------------+--------+
| NP01       | Prashant Phuyal  | LT01     |      10 | 2001-01-16 | MALE   |
| NP02       | Razz BC          | LT01     |      11 | 2001-02-11 | MALE   |
| NP03       | Aayush Shrestha  | LT01     |      12 | 2001-02-19 | MALE   |
| NP04       | Rachana Subedi   | LT02     |       9 | 2001-05-16 | FEMALE |
| NP05       | Aavinab Shah     | LT02     |      14 | 2001-07-11 | MALE   |
| NP06       | Shruti Kc        | LT02     |      16 | 2000-09-11 | FEMALE |
| NP07       | Sam Gautam       | LT03     |       1 | 2002-01-03 | MALE   |
| NP08       | Jeevan Risal     | LT03     |       5 | 2001-07-04 | MALE   |
| NP09       | Kareena Shrestha | LT04     |       7 | 2000-01-11 | FEMALE |
| NP10       | Babi Acharya     | LT04     |       5 | 2001-07-04 | FEMALE |
| NP11       | Prakriti Phuyal  | LT04     |      10 | 2002-01-02 | FEMALE |
+------------+------------------+----------+---------+------------+--------+
```

2. Write a query to update student staff name, work role, staff phone number from staff table of staff id SF06.

a. Solution

UPDATE `Staff`

SET

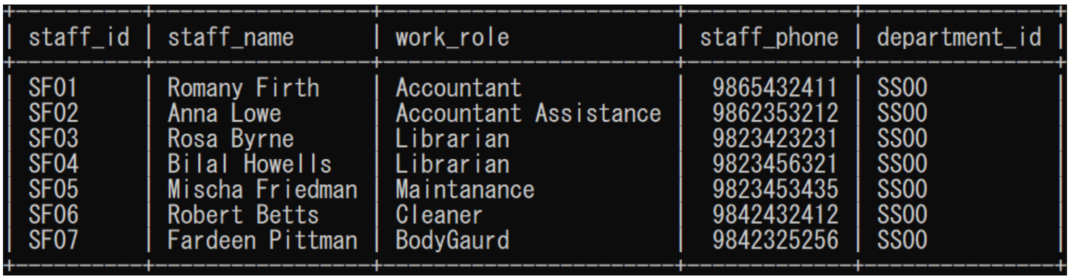
`staff_name` = 'Rose Khadka',

`work_role` = 'Event Manager',

`staff_phone` = '9842223333'

WHERE `staff_id` = 'SF06';

b. Result



```
+----------+-----------------+--------------------------+-------------+---------------+
| staff_id | staff_name      | work_role                | staff_phone | department_id |
+----------+-----------------+--------------------------+-------------+---------------+
| SF01     | Romany Firth    | Accountant               |  9865432411 | SS00          |
| SF02     | Anna Lowe       | Accountant Assistance    |  9862353212 | SS00          |
| SF03     | Rosa Byrne      | Librarian                |  9823423231 | SS00          |
| SF04     | Bilal Howells   | Librarian                |  9823456321 | SS00          |
| SF05     | Mischa Friedman | Maintanance              |  9823453435 | SS00          |
| SF06     | Robert Betts    | Cleaner                  |  9842432412 | SS00          |
| SF07     | Fardeen Pittman | BodyGaurd                |  9842325256 | SS00          |
+----------+-----------------+--------------------------+-------------+---------------+
```

3. Write a query to update teacher address and teacher department from teacher table of teacher having id 110

a. Solution

UPDATE `Teacher`

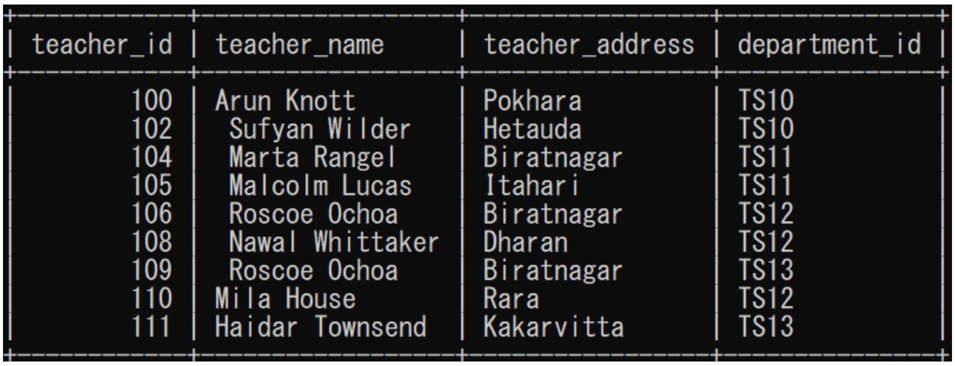
SET

`teacher_address` = 'Rara Lake',

`department_id` = 'TS10'

WHERE `teacher_id` = 110;

b. Result



```
+------------+-------------------+-----------------+---------------+
| teacher_id | teacher_name      | teacher_address | department_id |
+------------+-------------------+-----------------+---------------+
|        100 | Arun Knott        | Pokhara         | TS10          |
|        102 |  Sufyan Wilder    | Hetauda         | TS10          |
|        104 |  Marta Rangel     | Biratnagar      | TS11          |
|        105 |  Malcolm Lucas    | Itahari         | TS11          |
|        106 |  Roscoe Ochoa     | Biratnagar      | TS12          |
|        108 |  Nawal Whittaker  | Dharan          | TS12          |
|        109 |  Roscoe Ochoa     | Biratnagar      | TS13          |
|        110 | Mila House        | Rara            | TS12          |
|        111 | Haidar Townsend   | Kakarvitta      | TS13          |
+------------+-------------------+-----------------+---------------+
```

4. Write a query to update teacher subject with id 110 has to teach from table teacher_teaches.

a. Solution

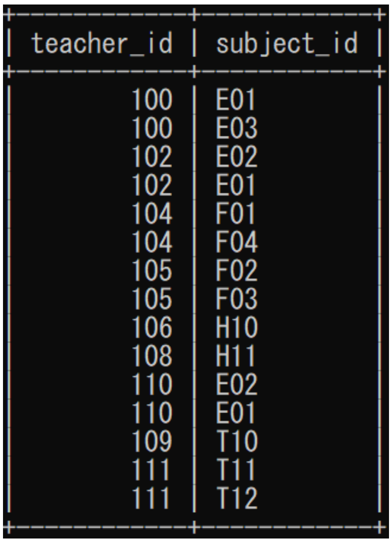
UPDATE `teacher_teaches`

SET

`subject_id` = 'E02'

WHERE teacher_id = '110'  Limit 1;

b. Result



```
+------------+------------+
| teacher_id | subject_id |
+------------+------------+
|        100 | E01        |
|        100 | E03        |
|        102 | E02        |
|        102 | E01        |
|        104 | F01        |
|        104 | F04        |
|        105 | F02        |
|        105 | F03        |
|        106 | H10        |
|        108 | H11        |
|        110 | E02        |
|        110 | E01        |
|        109 | T10        |
|        111 | T11        |
|        111 | T12        |
+------------+------------+
```

5. Write a query to update time period of subject H13 from table periodHour.

a. Solution

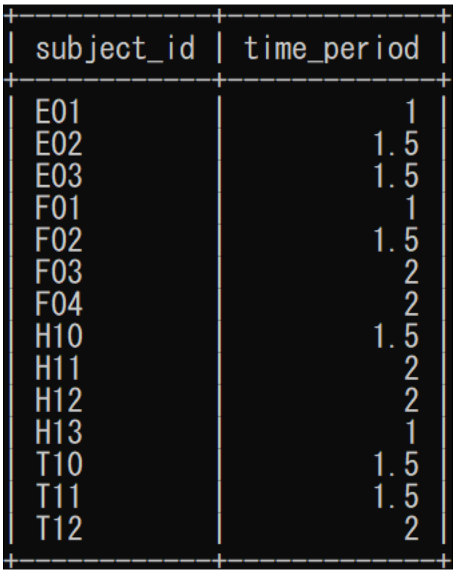
UPDATE `periodhour`

SET

`time_period` = 1

WHERE subject_id = 'H13';

b. Result



```
+------------+-------------+
| subject_id | time_period |
+------------+-------------+
| E01        |           1 |
| E02        |         1.5 |
| E03        |         1.5 |
| F01        |           1 |
| F02        |         1.5 |
| F03        |           2 |
| F04        |           2 |
| H10        |         1.5 |
| H11        |           2 |
| H12        |           2 |
| H13        |           1 |
| T10        |         1.5 |
| T11        |         1.5 |
| T12        |           2 |
+------------+-------------+
```

## XI. Delete Statement

1. Write a query to remove student NP11 from course.

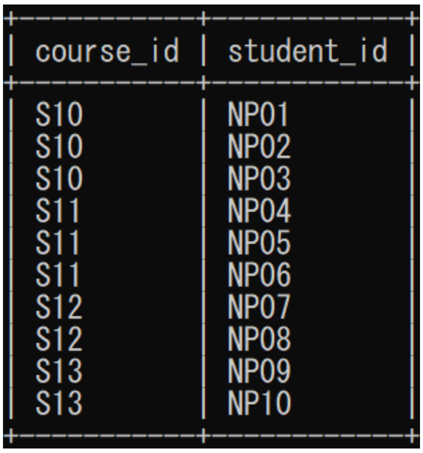
a. Solution

DELETE

FROM `student_course`

WHERE `student_id`= 'NP11';

b. Result



```
+-----------+------------+
| course_id | student_id |
+-----------+------------+
| S10       | NP01       |
| S10       | NP02       |
| S10       | NP03       |
| S11       | NP04       |
| S11       | NP05       |
| S11       | NP06       |
| S12       | NP07       |
| S12       | NP08       |
| S13       | NP09       |
| S13       | NP10       |
+-----------+------------+
```

2. Write a query to remove student NP11 library membership.

a. Solution

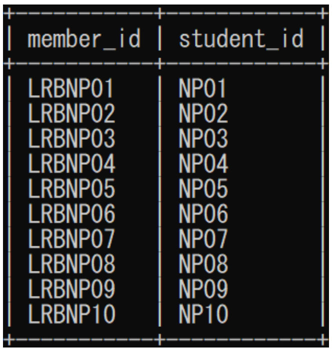
DELETE

FROM

`members`

WHERE `student_id`= 'NP11';

b. Result



```
+-----------+------------+
| member_id | student_id |
+-----------+------------+
| LRBNP01   | NP01       |
| LRBNP02   | NP02       |
| LRBNP03   | NP03       |
| LRBNP04   | NP04       |
| LRBNP05   | NP05       |
| LRBNP06   | NP06       |
| LRBNP07   | NP07       |
| LRBNP08   | NP08       |
| LRBNP09   | NP09       |
| LRBNP10   | NP10       |
+-----------+------------+
```

3. Write a query to delete parent data of student NP11 from parent table.

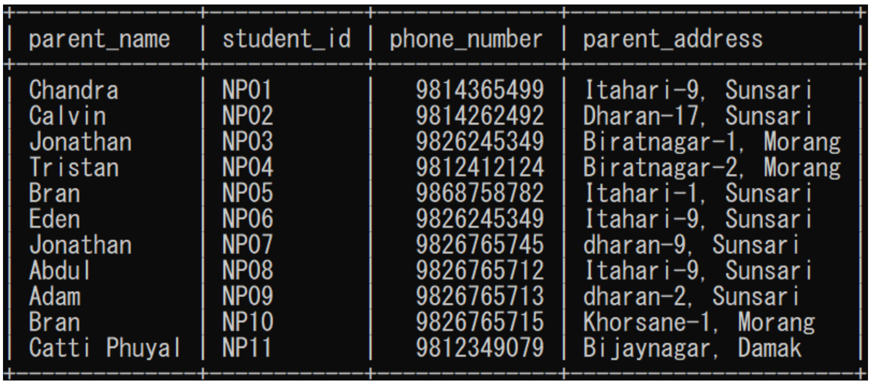
a. Solution

DELETE

FROM `Parent`

WHERE `student_id`= 'NP11';

b. Result



```
+-------------+------------+--------------+---------------------+
| parent_name | student_id | phone_number | parent_address      |
+-------------+------------+--------------+---------------------+
| Chandra     | NP01       |   9814365499 | Itahari-9, Sunsari  |
| Calvin      | NP02       |   9814262492 | Dharan-17, Sunsari  |
| Jonathan    | NP03       |   9826245349 | Biratnagar-1, Morang|
| Tristan     | NP04       |   9812412124 | Biratnagar-2, Morang|
| Bran        | NP05       |   9868758782 | Itahari-1, Sunsari  |
| Eden        | NP06       |   9826245349 | Itahari-9, Sunsari  |
| Jonathan    | NP07       |   9826765745 | dharan-9, Sunsari   |
| Abdul       | NP08       |   9826765712 | Itahari-9, Sunsari  |
| Adam        | NP09       |   9826765713 | dharan-2, Sunsari   |
| Bran        | NP10       |   9826765715 | Khorsane-1, Morang  |
| Catti Phuyal| NP11       |   9812349079 | Bijaynagar, Damak   |
+-------------+------------+--------------+---------------------+
```

4. Write a query to delete student data of student NP11 from student table.

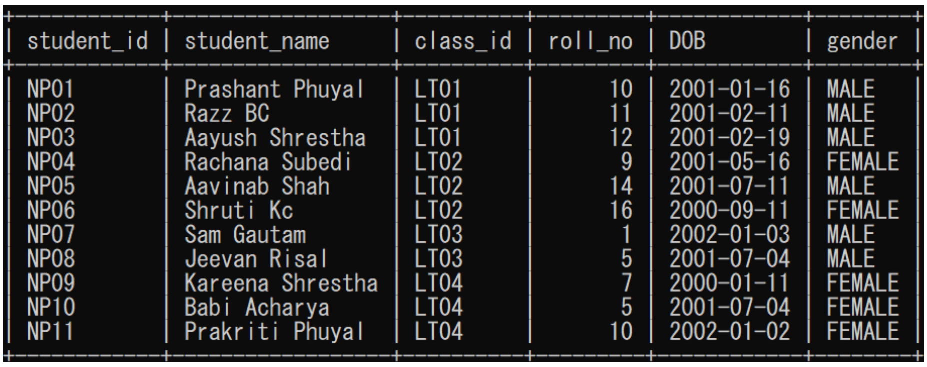
a. Solution

DELETE

FROM `Student`

WHERE `student_id`= 'NP11';

b. Result



```
+------------+-----------------+----------+---------+------------+--------+
| student_id | student_name    | class_id | roll_no | DOB        | gender |
+------------+-----------------+----------+---------+------------+--------+
| NP01       | Prashant Phuyal | LT01     |      10 | 2001-01-16 | MALE   |
| NP02       | Razz BC         | LT01     |      11 | 2001-02-11 | MALE   |
| NP03       | Aayush Shrestha | LT01     |      12 | 2001-02-19 | MALE   |
| NP04       | Rachana Subedi  | LT02     |       9 | 2001-05-16 | FEMALE |
| NP05       | Aavinab Shah    | LT02     |      14 | 2001-07-11 | MALE   |
| NP06       | Shruti Kc       | LT02     |      16 | 2000-09-11 | FEMALE |
| NP07       | Sam Gautam      | LT03     |       1 | 2002-01-03 | MALE   |
| NP08       | Jeevan Risal    | LT03     |       5 | 2001-07-04 | MALE   |
| NP09       | Kareena Shrestha| LT04     |       7 | 2000-01-11 | FEMALE |
| NP10       | Babi Acharya    | LT04     |       5 | 2001-07-04 | FEMALE |
| NP11       | Prakriti Phuyal | LT04     |      10 | 2002-01-02 | FEMALE |
+------------+-----------------+----------+---------+------------+--------+
```

5. Write a query to delete issued record of book having id 1006 from issued_book table.

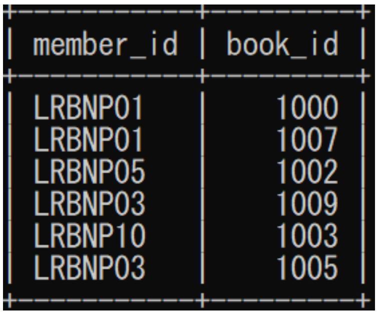
a. Solution

DELETE

FROM `issued_books`

WHERE `book_id`= 1006

b. Result



```
+-----------+---------+
| member_id | book_id |
+-----------+---------+
| LRBNP01   |    1000 |
| LRBNP01   |    1007 |
| LRBNP05   |    1002 |
| LRBNP03   |    1009 |
| LRBNP10   |    1003 |
| LRBNP03   |    1005 |
+-----------+---------+
```

# XII. Normalization

Database Normalization is the process of organizing data in the database. It is the systematic process of eliminating data redundancy and undesirable characteristics like Update, Insertion and Deletion Anomalies. If the data is not normalized and have data repetition then it would occupy more space and makes difficult in handling operations like deletion and insertion. The added advantage of getting an organized data enhances the performance level. The main purpose of normalization is:

Increased consistency: Database normalization decreases the repetition of data placing data in one place and one place only, reducing the possibility of inconsistent data.

Easier object-to-data mapping: Database maintained under normalization rules in general are conceptually closer to object-oriented schemas as the object-oriented achievements of enhancing high cohesion and loose coupling between classes results in similar solutions.

Normalization rules are divided into following normal forms:

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal From (3NF)

## A. First Normal Form

The first normal form expects to follow a few simple rules while designing the database:

- Each column should contain atomic values.
- A column should contain values that are of the same type.
- Each column should have a unique name.
- Order in which data is saved doesn’t matter.

### 1NF Example

Let us create one Student table with Student Name, Roll No and Subject as shown below:

| Rollno | name | Subject |
|---|---|---|
| 101 | Prashant | OS, CN |
| 103 | Aayush | JAVA |
| 102 | Aavinab | C, C++ |

Here out of the 3 student 2 has more than 1 subjects, and I have stored in them in single column but as per first normal form each table should contain atomic values, which leads to the violation of 1NF.

So, we can solve this problem by breaking the 2 values in single column to atomic values as shown below:

| Rollno | name | Subject |
|---|---|---|
| 101 | Prashant | OS |
| 101 | Prashant | CN |
| 103 | Aayush | JAVA |
| 102 | Aavinab | C |
| 102 | Aavinab | C++ |

(Study Tonight, 2020)

Although there is a repeatation of data in table, but values for subject column is atomic for each row.

### B. Second Normal Form

For a table to be in second normal form we should follow a few simple rules:

- It should be in 1st normal form
- It should not have any partial dependencies

Now, let us create a score table with attribute score id, student id, subject id, marks and teacher.

| Score_id | Student_id | Subject_id | marks | Teacher |
|---|---|---|---|---|
| 1 | 10 | 1 | 82 | Mr. OS |
| 2 | 10 | 2 | 77 | Mrs. CN |
| 3 | 11 | 1 | 85 | Mr. JAVA |
| 4 | 11 | 2 | 82 | Mr. C |
| 5 | 11 | 4 | 95 | Mr. C++ |

The table follows 1NF rules as each attribute has atomic values. However, it does not follows second normal form because teacher column depends on subject id but it has nothing to do with student which leads to partial dependency and violates the second normal form.

There are many different solution to remove partial dependency the only objective is to remove teacher form score table.

One of the way is to move the teacher column to the subject table where it fits appropriately along with the subject names.

| Subject_id | Subject name | Teacher |
|---|---|---|
| 1 | OS | Mr. OS |
| 2 | CN | Mrs. CN |
| 1 | JAVA | Mr. JAVA |
| 2 | C | Mr. C |
| 4 | C++ | Mr. C++ |

Also, we can create another separate table for the teacher and use the teacher's id where ever we want.

| Subject_id | Teacher |
|---|---|
| 1 | Mr. OS |
| 2 | Mrs. CN |
| 3 | Mr. JAVA |
| 4 | Mr. C |
| 5 | Mr. C++ |

(Study Tonight, 2020)

## C. Third Normal Form

For a table to be in third normal form we should follow a few simple rules:

- It should be in 2st normal form
- It should not have transitive dependencies

Now, let us create a score table with attribute score id, student_id, subject_id, marks and exam_name and total_marks.

| Score_id | Student_id | Subject_id | marks | Exam_name | Total_marks |
|---|---|---|---|---|---|
| 101 | 10 | 1 | 82 | OS | 120 |
| 102 | 10 | 2 | 77 | CN | 130 |
| 111 | 11 | 1 | 85 | JAVA | 110 |
| 112 | 11 | 2 | 82 | C | 140 |
| 114 | 11 | 4 | 95 | C++ | 120 |

Here, score_id is formed with the combination of two keys student_id and subject_id as a composite key. Hence, primary id is both student_id and subject_id and all other attributes depends on it expect total marks which depends on exam_name but exam name is not a part of primary key which leads to transitive dependency(when attribute on table depends on some non-prime attributes) which violates the third normal form.

Solution to this problem is we should take exam_name and total_marks and put them on exam table and use exam id where ever it requires.

| Exam_id | Exam_name | Total_marks |
|---|---|---|
| 1 | OS | 120 |
| 2 | CN | 130 |
| 3 | JAVA | 110 |
| 4 | C | 140 |
| 5 | C++ | 120 |

In this way, score table is in third normal form.

(Study Tonight, 2020)

## Conclusion

This project engaged me in understanding about database and SQL queries. The report contents successfully accomplished its achievement by allowing me to work with Xampp, MySQL Workbench and testing and learning SQL queries, which in turn helped me to understand database management system and further more. This coursework helped me to become more familiar with the concepts of database management, about its entities, attributes, ER diagram and queries. Syntax and functions to create, update and insert attributes were also a best experience. I felt Xampp as a complex shell for writing SQL queries so I used MySQL workbench first and after it I exported the script file from workbench and imported on xampp for further report screenshot purpose.

The coursework was an interesting experience that helped me learn the ways of database management systems and to operate them. I enjoyed a lot while doing this work although many problems were encountered throughout the coursework along with its process but they were quickly solved by the help respected teachers. I want to thank Mr. sachit tandukar sir, Mr. Subiran sir and Mr. Deepson sir for guide and support throughout the coursework. (w3schools.com, n.d.) (codeacademy, n.d.) (study tonight, n.d.)

## References

codeacademy. (n.d.). *List of SQL Commands*. Retrieved from code academy: https://www.codecademy.com/articles/sql-commands

Study Tonight. (2020). *studytonight*. Retrieved from Normalization of Database: https://www.studytonight.com/dbms/database-normalization.php

study tonight. (n.d.). *What are SQL Functions?* Retrieved from studytonight: https://www.studytonight.com/dbms/sql-function.php

w3schools.com. (n.d.). *mysql functions*. Retrieved from w3schools: https://www.w3schools.com/sql/sql_ref_mysql.asp