UNIVERSITY PARTNER
UNIVERSITY OF WOLVERHAMPTON
iAcademy
Quality Education Improving Lives

Academic Year	Module	Assessment Number	Assessment Type
S20	Database	A1	Individual Report

School Management System

Student Id : NP03A190017
Student Name : Roshan Parajuli
Section : DC3
Module Leader : Deepson Shrestha
Lecturer : Chiranjivi Khanal

Submitted on : 05-06-2020

Acknowledgement

With immense please I, Roshan Parajuli, presenting the database report on “School Management System” as a part of the curriculum of “BSc. (Hons) Computer Science” wish to thank all the people for giving me unending support while writing this report and for making it successful. As this report is an outcome of the constant inspiration and massive support of my Lecturer Chiranjivi Khanal and Module Leader Deepson Shrestha, I take this opportunity to express my profound gratitude and deep regards towards both of them. I would like to extend my thankfulness to the University of Wolverhampton as well as Herald College for providing the module contents and good cooperation during the whole semester. Last but not the least, I am thankful to my friends for playing a great role to help me complete this report.

Table of Contents

1. SYSTEM DESCRIPTION

Aim:

For any management, it is difficult for the manual or papery storage of a large amount of data. In this case, it is a challenge for the management to provide information immediately if any user wants it because it takes a long time. To eliminate this problem and significantly utilize the time and effort, this school management system database is created. The data operations that needed the investment of a lot of time just takes a few minutes/seconds while using this database management system.

Shikshya Academy is a private school, containing a number of staff including teachers, chefs, cleaners, and of course, students. Many students get admitted here every year and the storage of data in a file management system is being troublesome because of integrity, atomicity, security, redundancy, and inconsistency, it provides. That is why, the creation of this database aims to speed up the system, reduce errors, and provide integrity, atomicity, security & consistency to the data of the school.

Description:

The main objective of the school management system is to monitor the details about the school, students, classes, employees, registrations, and so on. It oversees all classes, programs, registries, students, and its fragments. The project is completely planned at the administrative level, and only the administrator guarantees access. It records all the details related to admissions, attendance, classes, exams, fees, library, results, canteen, staff, students, transportation, and subjects.

The admission table contains records about the admission date of the student along with their name and admission fee paid. The exam table contains the details about the examination date, time, subject as well the room where the examination is going to be conducted. The transportation table contains the details about the vehicles owned by the school including their number, capacity, route, mode, cost and the students for whom the vehicles are being utilized for. The two attendance tables i.e. student

attendance and staff attendance keep a record of the attendance details including total days, present days and absent days. The canteen table records the information about the food items along with their prices that are consumed by the students. The class table contains the records about the class id, name, capacity, and the location of those classes. The fee table contains the records about the total fee, due fee and paid fee of each and every student. The result table contains information about the full marks, pass marks, marks obtained as well as the grade of each student in each subject. The staff table contains the records about the staff id, name, designation, date of birth, and their communication medium like email and mobile number. The subject table contains records about each subject along with its id and description. Finally, the student table contains records about the student id, name, address, phone, email, date of birth, guardians name, and guardians phone number. The book table consists of the details like id, name, author, edition and quantity of all the book present in the library. The library table contains the information about the student who borrowed the book, the id of the book borrowed, borrowed date and the date to be returned.

2. DATA DICTIONARIES

1. Student

This table consists of details about the personal information of every student. The information stored in this table includes student id, name, address, mobile number, email address, date of birth and one of guardian's name & their phone number.

Constraint: Since student id will be unique for each student, it is used as a primary key. There is no presence of foreign key in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
P	student_id	Integer		11	Unique identifier for every student.	17
	student_name	String		40	The full name of the student.	Roshan Parajuli
	student_address	String		50	The temporary address of Student.	Dakshin Dhoka, Kathmandu
	student_mobile	Decimal	NNNNNNNNNN	10,0	The phone number of the student.	9841000678
	student_email	String		30	The email address of the student.	Roshan.parajuly1@gmail.com
	student_dob	Date	YYYY-MM-DD	10	The date of birth of student.	2000-03-18
	guardian_name	String		40	The full name of one of student's guardians.	Dhiraj Parajuli
	guardian_phone	Decimal	NNNNNNNNNN	10,0	The phone number of one of student's guardians.	9840378984

2. Class

This table consists of details about the all the classes of the school. The information stored in this table includes class id, class name, capacity and the block where it is situated in.

Constraint: Since class id will be unique for each and every class, it is used as a primary key. There is no presence of foreign key in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
P	class_id	String	XNN	3	Unique identifier for the class.	C01
	class_name	String		25	The name of the class.	Lecture Theatre 1
	location	String		25	The room where class is held.	Block A
	class_capacity	Integer		11	The capacity about the class.	70

3. Subject

This table consists of details about all the subjects that are taught within the school premises. The information stored in this table includes subject id, name and description.

Constraint: Since subject id will be unique for each and every subject of different grades, it is used as a primary key. There is no presence of foreign key in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
P	Subject_id	String	X00N	4	Unique identifier for the subject.	A1
	Subject_name	String		25	The name of the subject.	Mathematics
	Subject_description	String		50	The description about the subject.	Fundamental Principles of Calculus.

4. Staff

This table consists of details about the personal as well as academic information of every staff present within the school premises. The information stored in this table includes staff id, full name, job / designation, email address, date of birth and their phone number.

Constraint: Since staff id will be unique for each staff, it is used as a primary key. There is no precense of foreign key in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
P	staff_id	String	X0N	10	Unique identifier for the staffs.	S11
	staff_name	String		50	The full name of the staff.	Arunima Lamsal
	staff_job	String		25	The job / designation of staff.	Teacher
	staff_dob	Date	YYYY-MM-DD	10	The date of birth of staff.	2000-03-12
	staff_email	String		30	The email address of staff.	rsnpj1@gmail.com
	staff_mobile	Decimal	NNNNNNNNNN	10,0	The phone number of staff.	9840000090

5. Book

This table consists of details about the books in the library section. The information stored in this table includes book id, name, full name of the author, edition and the quantity of that book present in the library.

Constraint: Since book id will be unique for each and every book that belongs to the library, it is used as a primary key. There is no presence of foreign key in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
P	book_id	Integer		11	The unique identifier for every book	12
	book_name	String		20	The name of the book	Revolution 20-20
	author	String		30	The name of the author of the book	Chetan Bhagat
	edition	String		5	The edition of the book present.	3rd
	quantity	Int		11	The total quantity of same books present in library.	20

6. Library

This table consists of details about the library and the relationship between books and students. The information stored in this table includes student id of the student who borrowed the book, the id of the book borrowed, the date of borrow and the date where the book is destined to be returned.

Constraint: Here student id from student table and book id from book table act as foreign key. Primary key is not present in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
F	student_id	int		11	Student id from student table	17
F	book_id	Integer		11	The id of book borrowed from library.	1
	borrowed_date	Date	YYYY-MM-DD	10	The date when book was borrowed.	2077-03-18
	return_date	Date	YYYY-MM-DD	10	The date when book should be returned,	2077-04-15

7. Fee

This table consists of details about the payment of fee by the students. The information stored in this table includes bill number, student id, total fee, due fee and paid fee.

Constraint: Here, the bill number is unique for each transaction and that is why it is a primary key. Student id from student table is the foreign key.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
P	bill_number	String	X00N	5	Unique identifier for the bill.	f025
F	student_id	int		11	Student id from student table	17
	total_fee	Decimal		8,2	The total fee of the student.	10450.25
	fee_due	Decimal		8,2	The remaining fee of the student.	10500
	fee_paid	Decimal		8,2	The cleared fee by the student.	50000

8. Admission

This table consists of details about the admission of students. The information stored in this table includes admission id (or registration number), admission charges, student's name and the date of admission.

Constraint: Here admission id is unique for each and every new student who is admitted in the school. Foreign key is not present in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
P	admission_id	String	XX000N	10	Unique identifier for the admission.	AD1
	admission_fee	decimal		10	The fee that was deposited during admission.	17000.00
	student_name	String		50	The name of student registered during admission.	Ramlal Jyadav
	admission_date	Date	YYYY-MM-DD	10	The date of admission.	2077-04-15

9. Result

This table consists of details about the result of students. The information stored in this table includes student id, subject id, full marks, pass marks, marks obtained and total grade in a subject.

Constraint: Student id from student table and subject id from subject table are the two foreign keys in this table. Primary key is not present in here.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
F	student_id	Integer		11	Student id from student table.	17
F	subject_id	String		4	Subject id from subject table.	A2
	full_marks	Integer		11	Full marks of the subject.	100
	pass_marks	Integer		11	Pass marks of the subject.	40
	marks_obtained	Decimal	NN.NN	4,2	The total marks obtained by the student in the subject.	90.50
	grade	String		2	The grade of the subject.	A+

10. Transportation

This table consists of details about the transportation medium that are present within the school premises. The information stored in this table includes vehicle number, the route that it takes, its capacity, its type and monthly cost.

Constraint: Here vehicle number is unique for each and every vehicle and that is why it is referred to as a primary key. Foreign key is not present in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Examples
P	Vehicle_number	String		20	The unique number of the transportation mode.	Ba 1 Cha 1234
	Vehicle_route	String		50	The routes that the vehicle will cover.	Chabahil, Gaushala, Airport
	Vehile_capacity	Int		11	Maximum number of seats in the vehicle.	12
	Transportation_mode	String		20	The mode of transportation	Bus
	Transportation_cost	Decimal		6	Monthly cost of transportation	2000

11. Student transportation

This table consists of the linkage of student with any of the transportation medium that are present within the school premises. The information stored in this table includes student id and vehicle number.

Constraint: Student id from student table and vehicle number from transportation table are the two foreign keys. Primary key is not present in this table. Since, there are only two attributes and both of them are foreign keys, this table is a junction table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
F	Student_id	String		11	Unique identifier for the student.	10
F	Vehicle_number	String		20	The unique number of the transportation mode.	Ba 1 Cha 1234

12. Student attendance

This table consists of attendance records of every students. The information stored in this table includes student id, subject id, total days of the subject taught, present days and absent days.

Constraint: Student id from student table and subject id from subject table are the two foreign keys. Primary key is not present in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Examples
F	Student_id	Int		11	The unique identification number of students.	17
F	Subject_id	String	X000N	4	The unique identification number of subjects.	A1
	Total_days	Int		11	Total days till the end of a module.	120
	Present_days	int		11	Total present days of student throughout the end of the module.	110
	Absent_days	int		11	Total absent days of student throughout the end of the module.	10

13. Food

This table consists of details about all the food items that are available in the canteen. The name of the food and its price is included in this table.

Constraint: Since food name will be unique for each and every food, it is the primary key in this table. Foreign key is not present in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Examples
P	Food_name	String		20	The name of the food in the library.	Chicken momo
	Price	Integer		11	The price for each and every food item.	100

14. Canteen

This table consists of details on linkage between student table and food table. The information stored in this table includes student id, and lunch name.

Constraint: Student_id from student table and food name from food table are two foreign keys. There are no primary keys as this table is composed of two foreign keys only. It is also a junction table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
F	student_id	int		11	Student id from student table	17
F	food_name	String		20	The name of the lunch.	Chicken momo

15. Exam

This table consists of examination routine and further details about the same. The information stored in this table includes exam id, subject id, exam date, exam starting time, exam ending time and the class where the exam is going to be held at.

Constraint: Since, exam id will be unique for the table, it is the primary key in this table. Subject id from subject table and class id from class table are the two foreign keys.

Key	Field Name	Data Type	Data Format	Field Size	Description	Examples
P	Exam_id	int		11	The unique number of the column.	12
F	Subject_id	String		4	The unique identification number for each subject.	A12
	Exam_date	date	YYYY-MM-DD	10	Scheduled date of the examination.	2070-04-03
	Exam_start_at	time	HH: MM: SS	8	Scheduled starting time of the examination.	08:00:00
	Exam_end_at	time	HH: MM: SS	8	Scheduled ending time of the examination.	10:00:00
F	Exam_class_id	String		3	Class where the examination is held.	C10

16. Teacher

This table consists of details on linkage between staff table and subject table. The information stored in this table includes staff id, and subject id.

Constraint: Staff_id from staff table and subject_id from subject table are two foreign keys. There are no primary keys as this table is composed of two foreign keys only. It is also a junction table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
F	staff_id	String		10	Staff id from staff table	S15
F	Subject_id	String		4	The name of the subject from subject table.	A1

17. Staff attendance

This table consists of attendance details about all the staffs of the school. The information stored in this table includes staff id, total days, present days as well as absent days.

Constraint: Staff_id from staff table is a foreign key in this table. Primary key is not present in this table.

Key	Field Name	Data Type	Data Format	Field Size	Description	Example
P	staff_id	String	XNN	10	Unique identifier for the class.	S01
	Total_days	Integer		11	The name of the class.	90
	Present_days	Integer		11	The room where class is held.	80
	Absent_days	Integer		11	The capacity about the class.	10

3. ER DIAGRAM

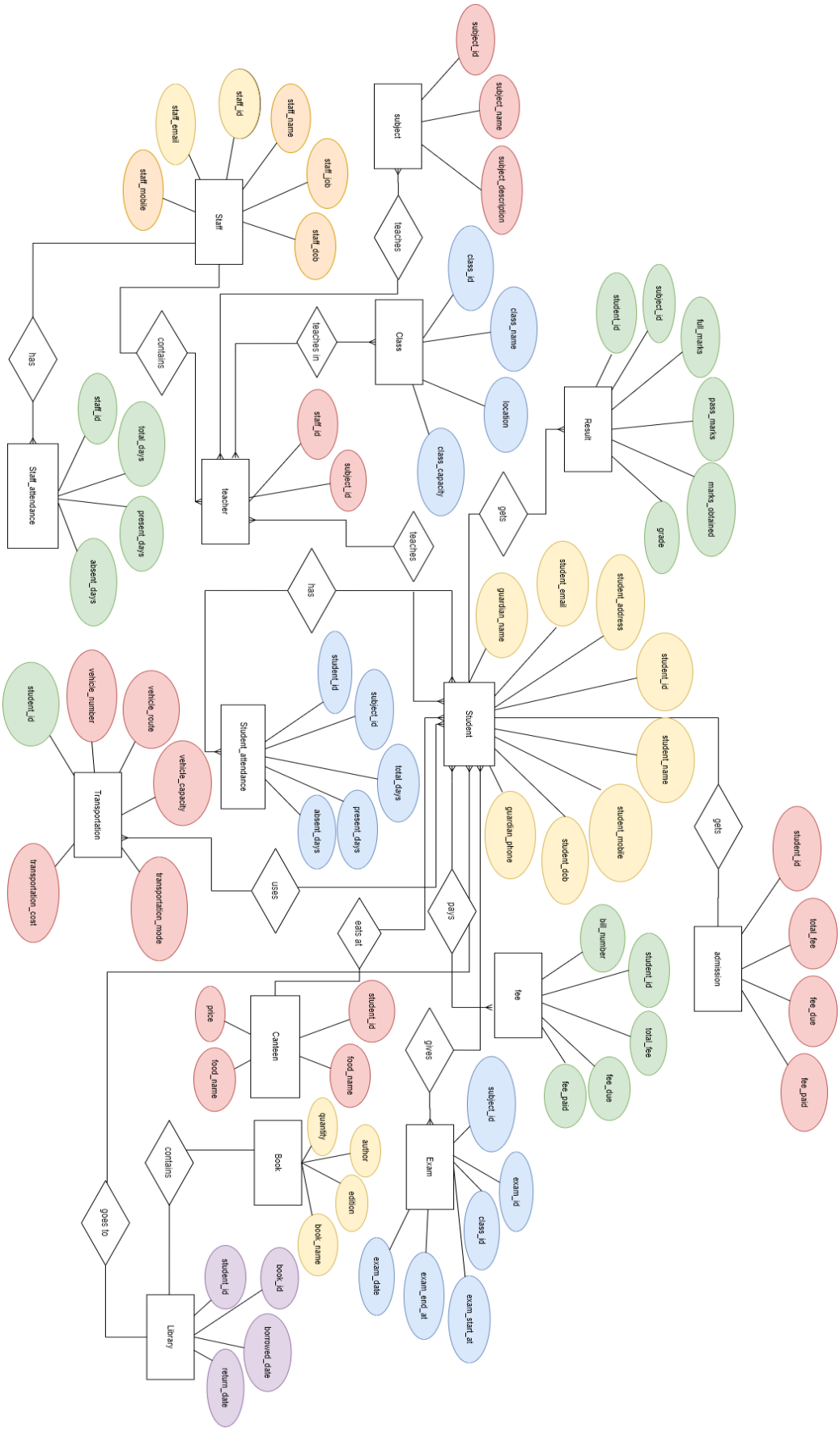


Figure 1: ER Diagram

4. RELATIONAL DATABASE SCHEMA

i. Student (**student_id**, student_name, student_address, student_mobile, student_email, student_dob, guardian_name, guardian_phone)

ii. Class (**class_id**, class_name, location, class_capacity)

iii. Subject (**subject_id**, subject_name, subject_description)

iv. Staff (**staff_id**, staff_name, staff_job, staff_dob, staff_email, staff_mobile)

v. Book (**book_id**, book_name, author, edition, quantity)

vi. Library (student_id, book_id, borrowed_date, return_date)

vii. Fee (**bill_number**, student_id, total_fee, fee_due, fee_paid)

viii. Admission (**admission_id**, admission_date, admission_fee, student_name)

ix. Result (student_id, subject_id, full_marks, pass_marks, marks_obtained, grade)

x. Transportation (**vehicle_number**, vehicle_route, vehicle_capacity, transportation_mode, transportation_cost)

xi. Student_transport (student_id, vehicle_number)

xii. Student_attendance (student_id, subject_id, total_days, present_days, absent_days)

xiii. Food (**food_name**, price)

xiv. Canteen (student_id, food_name)

xv. Exam (**exam_id**, subject_id, exam_date, exam_start_at, exam_end_at, class_id);

xvi. Teacher (staff_id, subject_id)

xvii. Staff_attendance (staff_id, total_days, present_days, absent_days)

Note: Primary keys are represented in **bold** and foreign keys are underlined.

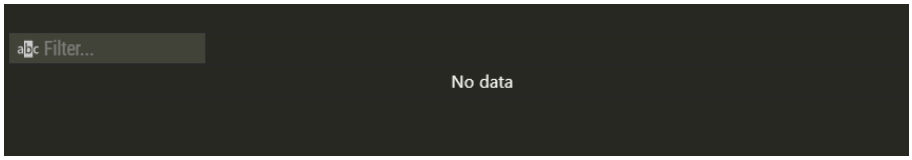
5. CREATING DATABASE

1. Creating database shikshya academy.

   Query:

```
create database shikshya_academy;
```

   Result:



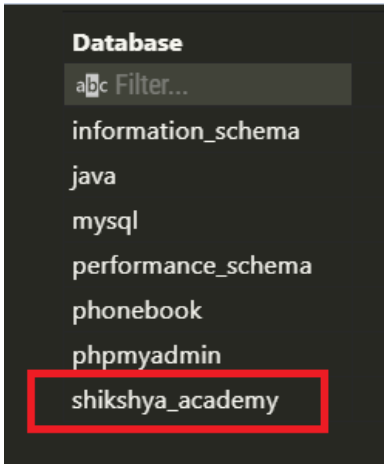
   Note: Here, Output window returns "No data" if Query was executed successfully without any errors.

2. Listing all databases.

   Query:

```
show databases;
```

   Result:

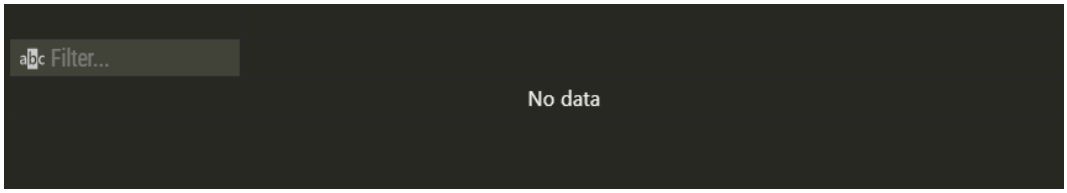


3. Using shikshya_academy database.

   Query:

```
use shikshya_academy;
```

   Result:



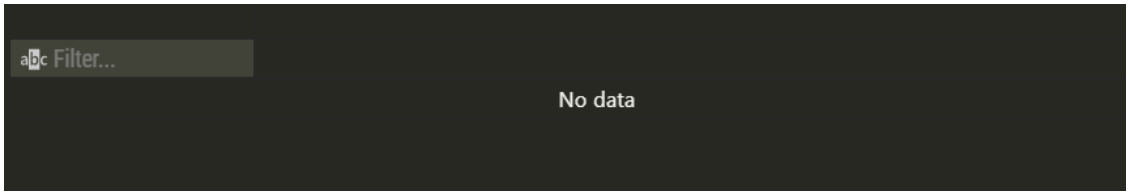
6. CREATING TABLES

1. Creating student table

   Query:

```
CREATE TABLE student (
  student_id int PRIMARY KEY,
  student_name varchar(40) NOT NULL,
  student_address varchar(50) ,
  student_mobile decimal(10, 0) ,
  student_email varchar(30) DEFAULT 'NULL',
  student_dob date ,
  guardian_name varchar(40) ,
  guardian_phone decimal(10, 0),
 CHECK (student_email LIKE "%@%")
);
```

   Result:



2. Describing student table.

   Query:

```
desc student;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
student_id	int(11)	NO	PRI	NULL	
student_name	varchar(40)	NO		NULL	
student_address	varchar(50)	YES		NULL	
student_mobile	decimal(10,0)	YES		NULL	
student_email	varchar(30)	YES		NULL	
student_dob	date	YES		NULL	
guardian_name	varchar(40)	YES		NULL	
guardian_phone	decimal(10,0)	YES		NULL	

3. Inserting values into student table.

   Query:

```
INSERT INTO student (
    student_id,
    student_name,
    student_address,
    student_mobile,
    student_email,
    student_dob,
    guardian_name,
    guardian_phone
  )
VALUES
  (
    1,
    'Roshan Parajuli',
    'Dakshin Dhoka',
    '9840378984',
    'roshan.parajuly1@gmail.com',
    '2057-03-18',
    'Dhirnath Parajuli',
    '9841992081'
  ),
```

```
  (
    2,
    'Prakriti Dhakal',
    'Naxal',
    '9840378986',
    'prakriti_mutu@gmail.com',
    '2059-05-19',
    'Uttam Dhakal',
    '9841902070'
  ),
  (
    3,
    'Billu Prasad',
    'Thali',
    '9840572954',
    'billubarker123@gmail.com',
    '2057-05-24',
    'Pankaj Prasad',
    '9841152289'
  ),
  (
    4,
    'Maya Kafle',
    'Narayantar',
    '9828563476',
    'kaflemaya144@hotmail.com',
    '2058-09-01',
    'Bhola Kafle',
    '9825678423'
  ),
```

```
  (
    5,
    'Rita Dangol',
    'Gokarna',
    '9841563686',
    'rita.rita1@hotmail.com',
    '2057-04-04',
    'Dibesh Dangol',
    '9825678423'
  ),
  (
    6,
    'Dristi Sigdel',
    'Chabahil',
    '9841758920',
    'ilovemyself@hotmail.com',
    '2057-06-12',
    'Sparsha Sigdel',
    '9842768940'
  ),
  (
    7,
    'Sugandhi Chaudhary',
    'Sankhu',
    '9841898979',
    'sugandhi.me@gmail.com',
    '2057-05-10',
    'Mahesh Chaudhary',
    '9842767690'
  ),
```

```
  (
    8,
    'Saumya Bhandari',
    'Changunarayan',
    '9841850974',
    'olafmaximus@gmail.com',
    '2058-05-02',
    'Aadarsha Bhandari',
    '9842647484'
  ),
  (
    9,
    'Aayush Shrestha',
    'Pashupati',
    '9841573980',
    'shresthaaaa@gmail.com',
    '2057-01-01',
    'Mayush Shrestha',
    '9841654983'
  ),
  (
    10,
    'Arjun Bhat',
    'Bhaisepati',
    '9840787878',
    'arjun1@gmail.com',
    '2059-12-12',
    'Mahesh Bhat',
    '9840378984'
  ),
```

```
  (
    11,
    'Nikhil Upreti',
    'Sundhara',
    '9830123459',
    'nikhil12@gmail.com',
    '2055-06-12',
    'Rajesh Upreti',
    '9845452345'
  ),
  (
    12,
    'Bijay Dangal',
    'Dakshin Dhoka',
    '9854670909',
    'bizzay1@yahoo.com',
    '2057-05-18',
    'Shanta Dangal',
    '9856453456'
  ),
  (
    13,
    'Abishek Shrestha',
    'Makalbari',
    '9856049586',
    'abishek.sh@yahoo.com',
    '2052-05-04',
    'Pasang Shrestha',
    '9845356739'
  ),
```

```
  (
    14,
    'Sandesh Khatiwada',
    'Gagalphedi',
    '9812345678',
    'sandy1@gmail.com',
    '2056-09-18',
    'Prem Khatiwada',
    '9856356789'
  ),
  (
    15,
    'Saman Tamrakar',
    'Dhobighat',
    '9842567040',
    'tamrasaman@yahoo.com',
    '2057-04-04',
    'Sahesh Tamrakar',
    '9841567832'
  ),
  (
    16,
    'Rohan Dahal',
    'Kamaladi',
    '9804569876',
    'dahalrohan21@yahoo.com',
    '2057-05-24',
    'Aashutosh Dahal',
    '9807458469'
  ),
```

```
  (
    17,
    'Suman Khanal',
    'Narayantar',
    '9804567686',
    'khanal.suman1@yahoo.com',
    '2057-07-22',
    'Mayanta Khanal',
    '9840676744'
  ),
  (
    18,
    'Saugat Acharya',
    'Dhobighat',
    '9807777777',
    'charyafromdhobi1@yahoo.com',
    '2056-03-19',
    'Sumanta Acharya',
    '9840663057'
  ),
  (
    19,
    'Rohit Parajuli',
    'Narayantar',
    '9841663936',
    'rohitpj1@yahoo.com',
    '2057-08-26',
    'Dhiraj Parajuli',
    '9806764578'
  ),
```

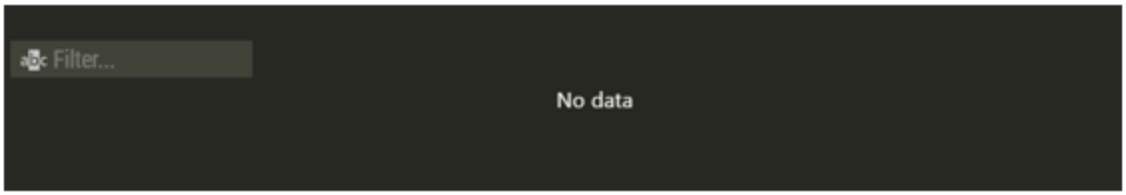
```
  (
    20,
    'Sumanta Dangal',
    'Aryalgaun',
    '9840768498',
    'sumantadangal@yahoo.com',
    '2059-02-12',
    'Prabhu Dangal',
    '9804509628'
  ),
  (
    21,
    'Urjit Bhattarai',
    'Gothatar',
    '9841465709',
    'urjit1@gmail.com',
    '2057-04-05',
    'Mahesh Bhattarai',
    '9804573956'
  ),
  (
    22,
    'Sanam Dulal',
    'Narayantar',
    '9841000032',
    'iamsanam1@gmail.com',
    '2057-06-09',
    'Sumanta Dulal',
    '9804444440'
  ),
```

```
  (
    23,
    'Pukar Dulal',
    'Narayantar',
    '9841000167',
    'iampukar1@gmail.com',
    '2057-06-09',
    'Sumanta Dulal',
    '9804444440'
  ),
  (
    24,
    'Ranbir Dulal',
    'Narayantar',
    '9841000343',
    'iamranbit1@gmail.com',
    '2057-06-09',
    'Sumanta Dulal',
    '9804444440'
  ),
  (
    25,
    'Ishan Banjara',
    'Narayantar',
    '9841674648',
    'banjaraish@gmail.com',
    '2057-04-09',
    'Hemanta Banjara',
    '9809090467'
  ),
```

```
  (
    26,
    'Arun Dahal',
    'Dakshin Dhoka',
    '9841111780',
    'arunman7@gmail.com',
    '2058-03-08',
    'Monica Dahal',
    '9805634521'
  ),
  (
    27,
    'Sukhadev Acharya',
    'Baneshwor',
    '9841888842',
    'sukhadevman@gmail.com',
    '2057-06-23',
    'Suyog Acharya',
    '9804444660'
  ),
  (
    28,
    'Suyog Baskota',
    'Narayantar',
    '9841008979',
    'baskota909@gmail.com',
    '2057-03-09',
    'Sangita Baskota',
    '9818345632'
  ),
```

```
(
  29,
  'Sheejan Satyal',
  'Narayantar',
  '9841123212',
  'iamssatyal@yahoo.com',
  '2057-02-28',
  'Prabin Satyal',
  '9806565789'
),
(
  30,
  'Arjun Parajuli',
  'Sankhu',
  '9841003321',
  'arjpj12@gmail.com',
  '2057-09-25',
  'Rukmina Parajuli',
  '9804567833'
);
```

Result:

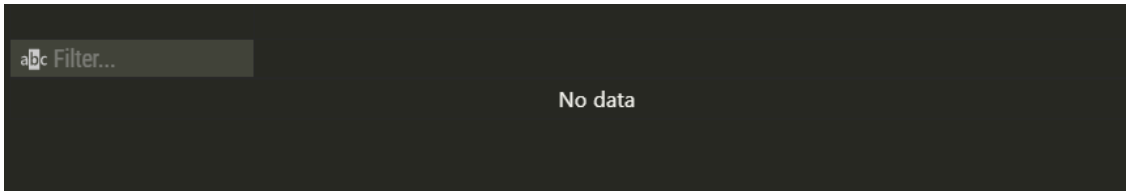


1. Creating class table

   Query:

```
CREATE TABLE class (
  class_id varchar(3) PRIMARY KEY,
  class_name varchar(25) ,
  location varchar(25) ,
  class_capacity int(11)
);
```

   Result:



2. Describing class table.

   Query:

```
desc class;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
class_id	varchar(3)	NO	PRI	NULL	
class_name	varchar(25)	YES		NULL	
class_room	varchar(25)	YES		NULL	
class_capacity	int(11)	YES		NULL	

3. Inserting values into class table.

   Query:

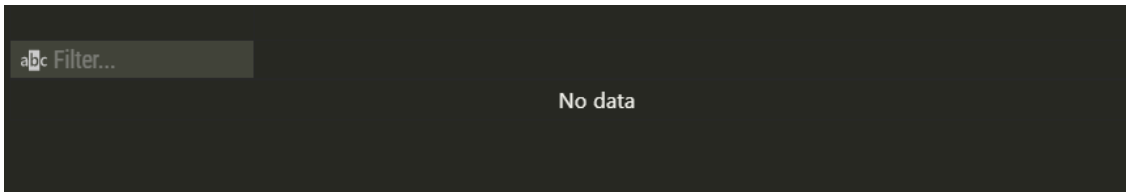
```
INSERT INTO class (
    class_id,
    class_name,
    location,
    class_capacity
  )
VALUES
  ('C01', 'Lecture Theatre 1', 'Block A', 70),
  ('C02', 'Lecture Theatre 2', 'Block A', 70),
  ('C03', 'Tutorial 1', 'Block B', 30),
  ('C04', 'Tutorial 2', 'Block B', 30),
  ('C05', 'Tutorial 3', 'Block A', 30),
  ('C06', 'Tutorial 4', 'Block A', 30),
  ('C07', 'Hall', 'Block B', 200),
  ('C08', 'Mega Hall', 'Block C', 300),
  ('C09', 'Computer Lab 1', 'Block B', 20),
  ('C10', 'Computer Lab 2', 'Block C', 30),
  ('C11', 'Lecture Theatre 3', 'Block D', 50),
  ('C12', 'Lecture Theatre 4', 'Block A', 70),
  ('C13', 'Lecture Theatre 5', 'Block A', 70),
  ('C14', 'Computer Lab 3', 'Block B', 30),
  ('C15', 'Computer Lab 4', 'Block A', 30),
  ('C16', 'Practical Room 1', 'Block A', 40),
  ('C17', 'Practical Room 2', 'Block B', 40),
  ('C18', 'Practical Room 3', 'Block C', 40),
  ('C19', 'Practical Room 4', 'Block D', 40),
  ('C20', 'Practical Room 5', 'Block D', 40);
```

1. Creating subject table

   Query:

```
CREATE TABLE subject (
  subject_id varchar(4) PRIMARY KEY,
  subject_name varchar(25) ,
  subject_description varchar(50) DEFAULT NULL
);
```

   Result:



2. Describing subject table.

   Query:

```
desc subject;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
subject_id	varchar(4)	NO	PRI	NULL	
subject_name	varchar(25)	YES		NULL	
subject_description	varchar(50)	YES		NULL	

3. Inserting values into subject table.

   Query:

```
INSERT INTO subject (
    subject_id,
    subject_name,
```

```
    subject_description
  )
VALUES
  (
    'A1',
    'Computer Science',
    'Basic concepts of computer science'
  ),
  (
    'A10',
    'Nepali',
    'Nepali stories and Byakaran'
  ),
  (
    'A11',
    'Moral Science',
    'Valuable lessons through stories'
  ),
  (
    'A12',
    'GK',
    'General Knowledge of important dates and
     events'
  ),
  (
    'A13',
    'Accounts',
    'Basic Principles of Accounting'
  ),
  (
```

```
    'A14',
    'History',
    'Brief history of Nepal'
  ),
  (
    'A15',
    'English Grammar',
    'Basic English grammar'
  ),
  (
    'A16',
    'Nepali Vyakaran',
    'Basic Nepali grammar'
  ),
  (
    'A17',
    'EPH',
    'Health, Population and Environment'
  ),
  (
    'A18',
    'Personal Development',
    'The development of ones character'
  ),
  (
    'A19',
    'Political Science',
    'History and present condition of society'
  ),
  (
```

```
    'A2',
    'Social studies ',
    'Study of society and politics'
  ),
  (
    'A20',
    'Physical Education',
    'Developing physical fitness'
  ),
  (
    'A21',
    'Marketing',
    'Basics of business and marketing'
  ),
  (
    'A22',
    'Economics',
    'Social science about the goods & services'
  ),
  (
    'A23',
    'Geography',
    'Relationships between people and their
     environment'
  ),
  (
    'A24',
    'Advanced Maths',
    'Pre Calculus and Algebra'
  ),
```

```
  (
    'A25',
    'Home Science',
    'Science related to home'
  ),
  (
    'A26',
    'Sahitya',
    'Study of Nepali literature'
  ),
  (
    'A27',
    'Advanced English',
    'Improve Reading and writing skills in
     English'
  ),
  (
    'A28',
    'Journalism',
    'Specialization of mass communication'
  ),
  (
    'A29',
    'Database',
    'Basics of MYSQL'),
  (
    'A3',
    'Mathematics',
    'Linear Algebra and Logic'
  ),
```

```
  (
    'A30',
    'Computer Architecture',
    'Basics of microprocessor'
  ),
  (
    'A4',
    'English',
    'Developing English writing skills'
  ),
  (
    'A5',
    'Programming',
    'Basic concepts of C Programming'
  ),
  (
    'A6',
    'Literature',
    'English poetry and novels'
  ),
  (
    'A7',
    'Physics',
    'Basic concepts of Physics'
  ),
  (
    'A8',
    'Chemistry',
    'Basic concepts of chemistry and reactions'
  ),
```

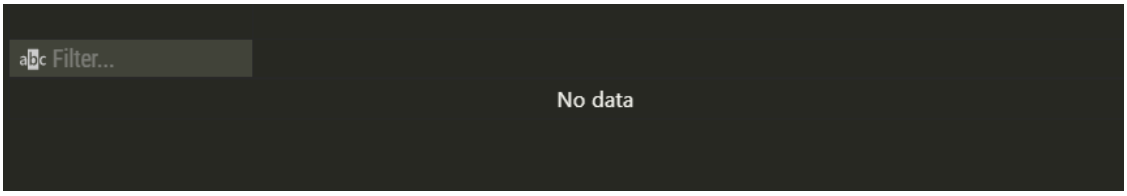
```
(
  'A9',
  'Opt. Mathematics',
  'Advanced concepts of mathematics'
);
```

1. Creating staff table

   Query:

```
CREATE TABLE staff (
  staff_id varchar(10) PRIMARY KEY,
  staff_name varchar(50),
  staff_job varchar(25),
  staff_dob date,
  staff_email varchar(30),
  staff_mobile decimal(10, 0)
)
```

   Result:



2. Describing staff table

   Query:

```
desc staff;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
staff_id	varchar(10)	NO	PRI	NULL	
staff_name	varchar(50)	YES		NULL	
staff_job	varchar(25)	YES		NULL	
staff_dob	date	YES		NULL	
staff_email	varchar(30)	YES		NULL	
staff_mobile	decimal(10,0)	YES		NULL	

3. Inserting values into staff table.

   Query:

```
INSERT INTO staff (
    staff_id,
    staff_name,
    staff_job,
    staff_dob,
    staff_email,
    staff_mobile
  )
VALUES
  (
    'S01',
    'Sudan Bhakta Acharya',
    'Teacher',
    '2040-03-02',
    'sudanbhakta@gmail.com',
    '9840788984'
  ),
  (
    'S02',
    'Sangeet Baskota',
    'Teacher',
    '2039-03-04',
    'baskotasan@gmail.com',
    '9840378984'
  ),
  (
    'S03',
    'Roshan Pyakurel',
    'Teacher',
    '2051-03-04',
```

```
    'pyakurelrosh1@gmail.com',
    '9840378790'
  ),
  (
    'S04',
    'Gajendra Kharel',
    'Cleaner',
    '2041-03-04',
    NULL,
    '9851345678'
  ),
  (
    'S05',
    'Ganesh Bhattarai',
    'Teacher',
    '2051-09-14',
    'ganeshbhatt21@gmail.com',
    '9849134523'
  ),
  (
    'S06',
    'Jiban Pokhrel',
    'Accountant',
    '2055-08-04',
    'jiban.pokhrel@gmail.com',
    '9849123456'
  ),
  (
    'S07',
    'Nanu Parajuli',
```

```
    'Principal',
    '2053-03-04',
    'Nanubhattarai1@gmail.com',
    '9840378790'
  ),
  (
    'S08',
    'Ramesh Lal Yadav',
    'Teacher',
    '2056-03-04',
    'ramesh1@gmail.com',
    '9840378745'
  ),
  (
    'S09',
    'Laxman Adhikari',
    'Watchman',
    '2056-08-04',
    'laxman.ad12@gmail.com',
    '9840678799'
  ),
  (
    'S10',
    'Manisha Bhattarai',
    'Teacher',
    '2047-06-02',
    'manisha.sha@gmail.com',
    '9840378456'
  ),
  (
```

```
    'S11',
    'Samyog Brahmachari',
    'Teacher',
    '2045-04-30',
    'samyog.yogsam@gmail.com',
    '9840378670'
  ),
  (
    'S12',
    'Ramesh Maan',
    'Teacher',
    '2050-12-19',
    'ramukaka12@gmail.com',
    '9840378778'
  ),
  (
    'S13',
    'Pankaj Bihari',
    'Teacher',
    '2043-12-04',
    'pakajbhaiya1@gmail.com',
    '9840376740'
  ),
  (
    'S14',
    'Ashish Gautam',
    'Teacher',
    '2057-02-14',
    'ashish.g@gmail.com',
    '9840378767'
```

```
),
(
  'S15',
  'Rahul Dulal',
  'Chef',
  '2051-05-14',
  NULL,
  '9840378712'
),
(
  'S16',
  'Sashwat Bhandari',
  'Teacher',
  '2047-08-13',
  'sassh1@gmail.com',
  '9840378090'
),
(
  'S17',
  'Abninav Bahadur',
  'Teacher',
  '2047-05-13',
  'abhinav1@gmail.com',
  '9840378089'
),
(
  'S18',
  'Ram Prasad Yadav',
  'Teacher',
  '2048-08-19',
```

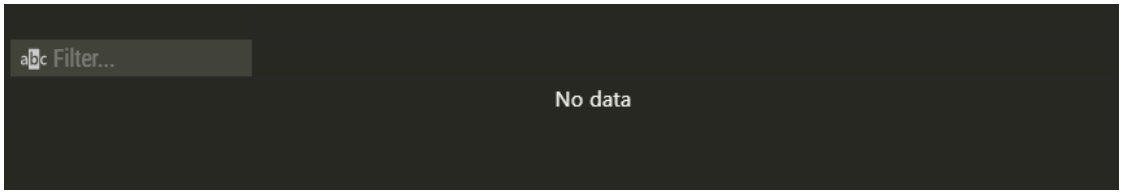
```
    'ramp1234@gmail.com',
    '9840378066'
  ),
  (
    'S19',
    'Hari Kumar',
    'Teacher',
    '2037-12-13',
    NULL,
    '9840378090'
  ),
  (
    'S20',
    'Aakash Shah',
    'Chef',
    '2057-09-13',
    NULL,
    NULL
  ),
  (
    'S21',
    'Minakshat Sharma',
    'Driver',
    '2047-09-13',
    'minak@gmail.com',
    '9840378765'
  ),
  (
    'S22',
    'Ram Kharel',
```

```
    'Driver',
    '2047-08-13',
    'ramkj12@gmail.com',
    '9840378666'
  ),
  (
    'S23',
    'Abishek Upmanyu',
    'Teacher',
    '2056-07-07',
    'abish1@yahoo.com',
    '9840123488'
  ),
  (
    'S24',
    'Shaleshma Shah',
    'Cleaning Lady',
    '2047-09-13',
    'singhshal1@gmail.com',
    '9841378095'
  ),
  (
    'S25',
    'Roshan Bhardwaj',
    'Warden',
    '2047-05-12',
    'bhardwajr1@gmail.com',
    '9840567834'
  ),
  (
```

```
    'S26',
    'Deepa Upadhyaye',
    'Teacher',
    '2045-09-08',
    'deepa.uph@yahoo.com',
    '9805630968'
  ),
  (
    'S27',
    'Junu Hada',
    'Teacher',
    '2046-06-06',
    'hadajunu1@yahoo.com',
    '9805653890'
  ),
  (
    'S28',
    'Arun Basyal',
    'Teacher',
    '2044-02-08',
    'arunbas1@yahoo.com',
    '9808888889'
  ),
  (
    'S29',
    'Suyog Amatya',
    'Teacher',
    '2047-08-08',
    'suyog.amatya1@yahoo.com',
    '9801233211'
```

```
),
(
  'S30',
  'Deepa Kuikel',
  'Teacher',
  '2053-08-27',
  'deepa.kuikel21@yahoo.com',
  '9807858034'
);
```

Result:

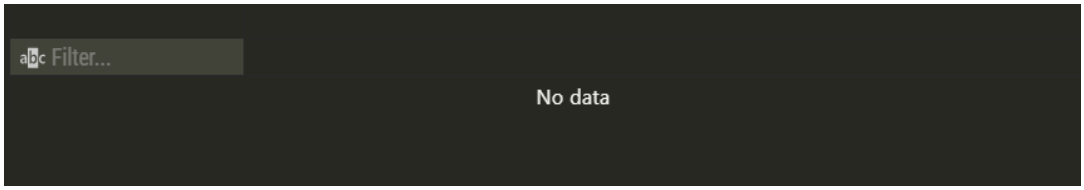


1. Creating book table

   Query:

```
CREATE TABLE book (
  book_id int(11) PRIMARY KEY,
  book_name varchar(20),
  author varchar(30),
  edition varchar(5),
  quantity int(11)
);
```

   Result:



2. Describing book table

   Query:

```
desc book;
```

Result:

Field	Type	Null	Key	Default	Extra
book_id	int(11)	NO	PRI	NULL	
book_name	varchar(20)	YES		NULL	
author	varchar(30)	YES		NULL	
edition	varchar(5)	YES		NULL	
quantity	int(11)	YES		NULL	

3. Inserting values into staff table.

Query:

```
INSERT INTO book (
    book_id,
    book_name,
    author,
    edition,
    quantity
  )
VALUES
  (
    1,
    '2 states',
    'Chetan Bhagat',
    '2nd',
    20
  ),
  (
    2,
    'The Three Musketeers',
    'Alexander Dumas',
    '1st',
    10
  ),
  (
    3,
    'Development',
    'Amartya Sen',
    '1st',
    10
  ),
  (
```

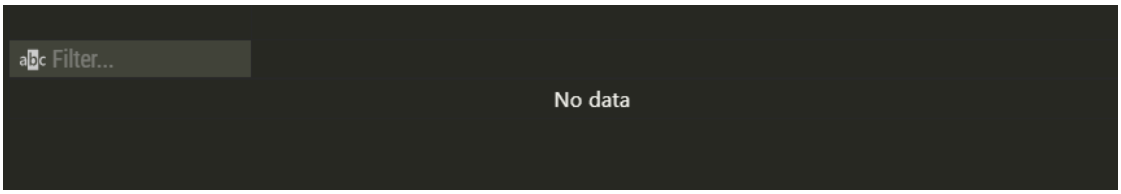
```
    4,
    'Clear City',
    'Anita Desai',
    '2nd',
    2
  ),
  (
    5,
    'Other Musketeers',
    'Alexander Dumas',
    '1st',
    10
  ),
  (
    6,
    'Circle of Reason ',
    'Amitav Ghose',
    '2nd',
    20
  ),
  (
    7,
    'Death of City',
    'Amrita Pritam',
    '2nd',
    20
  ),
  (
    8,
    'Politics',
```

```
    'Aristotle',
    '1st',
    10
  ),
  (
    9,
    'A Secular Agenda',
    'Arun Shourie',
    '1st',
    10
  ),
  (
    10,
    'Algebra',
    'Arundati Roy',
    '1st',
    10
  ),
  (
    11,
    'Bhagwat Geeta',
    'Ved Vyas',
    '1st',
    10
  ),
  (
    12,
    'Magic Seeds',
    'VS Naipaul',
    '1st',
```

```
    10
  ),
  (
    13,
    'The test of my Life',
    'Yuvraj Singh',
    '4th',
    30
  ),
  (
    14,
    'Against the Day',
    'Thomas Pychon',
    '1st',
    10
  ),
  (
    15,
    'Saya',
    'Subin Bhattarai',
    '1st',
    5
  ),
  (
    16,
    'Summer Love',
    'Subin Bhattarai',
    '1st',
    5
  ),
```

```
  (
    17,
    'Monsoon',
    'Subin Bhattarai',
    '1st',
    10
  ),
  (
    18,
    '2 states',
    'Chetan Bhagat',
    '1st',
    5
  ),
  (
    19,
    'Night at Call Center',
    'Chetan Bhagat',
    '2nd',
    15
  ),
  (
    20,
    'Half Girlfriend',
    'Chetan Bhagat',
    '1st',
    10
  );
```

Result:

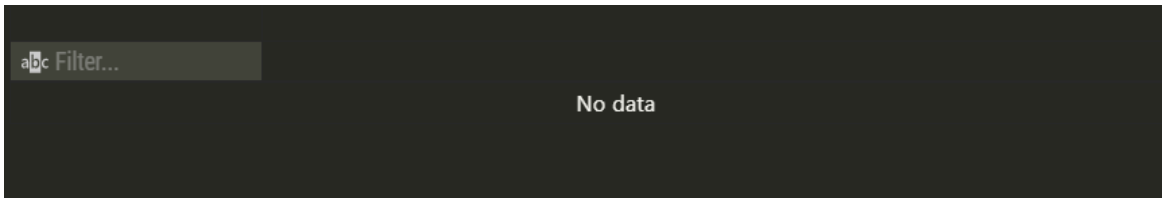


1. Creating library table

Query:

```
CREATE TABLE library (
  student_id int(11),
  borrowed_date date,
  return_date date,
  book_id int(11),
  FOREIGN KEY (student_id) REFERENCES student(
  student_id),
  FOREIGN KEY (book_id) REFERENCES book(book_id)
)
```

Result:



2. Describing library table

Query:

```
desc library;
```

Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
student_id	int(11)	YES	MUL	NULL	
borrowed_date	date	YES		NULL	
return_date	date	YES		NULL	
book_id	int(11)	YES	MUL	NULL	

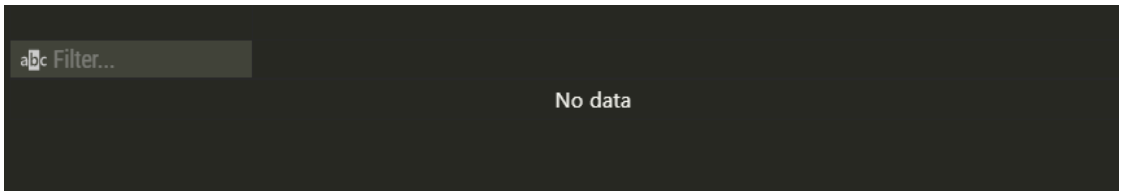
3. Inserting values into library table.

Query:

```
INSERT INTO library (
    student_id,
    borrowed_date,
    return_date,
    book_id
  )
VALUES
  (1, '2077-01-05', '2077-01-10', 1),
  (2, '2077-01-05', '2077-01-10', 1),
  (4, '2077-01-06', '2077-01-11', 1),
  (5, '2077-01-07', '2077-01-12', 1),
  (6, '2077-02-15', '2077-02-20', 3),
  (7, '2077-03-05', '2077-03-10', 3),
  (9, '2077-04-06', '2077-04-11', 4),
  (11, '2077-04-06', '2077-04-11', 4),
  (12, '2077-05-06', '2077-05-11', 1),
  (13, '2077-06-08', '2077-06-14', 1),
  (14, '2077-08-06', '2077-08-11', 1),
  (3, '2077-01-07', '2077-01-17', 1),
  (8, '2077-03-06', '2077-03-16', 1),
  (10, '2077-04-06', '2077-04-16', 4),
  (15, '2077-08-06', '2077-08-16', 5),
  (16, '2077-08-18', '2077-08-28', 2),
  (17, '2077-08-18', '2077-08-28', 2),
  (18, '2077-08-19', '2077-08-29', 2),
  (19, '2077-09-01', '2077-09-11', 2),
  (20, '2077-09-01', '2077-09-11', 2),
  (22, '2077-09-02', '2077-09-12', 2),
  (25, '2077-09-03', '2077-09-13', 2),
  (29, '2077-09-03', '2077-09-13', 2),
```

```
(30, '2077-09-07', '2077-09-17', 5);
```

Result:

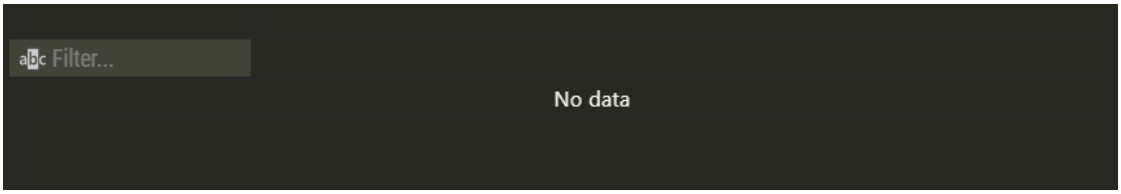


1. Creating fee table

   Query:

```
CREATE TABLE fee (
  bill_number varchar(5) PRIMARY KEY,
  student_id int NOT NULL,
  total_fee decimal(8, 2),
  fee_due decimal(8, 2),
  fee_paid varchar(1),
  FOREIGN KEY (student_id) REFERENCES student(
  student_id)
);
```

   Result:



2. Describing fee table

   Query:

```
desc fee;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
bill_number	varchar(5)	NO	PRI	NULL	
student_id	int(11)	YES	MUL	NULL	
total_fee	decimal(8,2)	YES		NULL	
fee_due	decimal(8,2)	YES		NULL	
fee_paid	decimal(8,2)	NO		NULL	

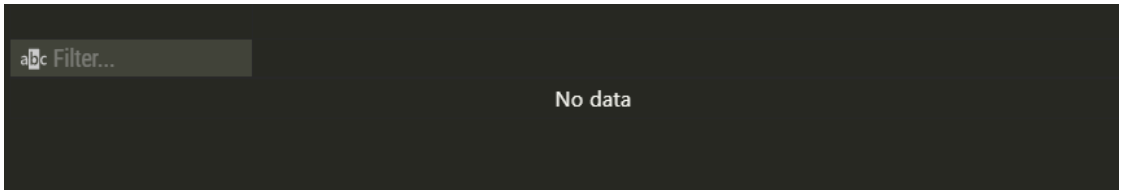
3. Inserting values into fee table.

Query:

```
INSERT INTO fee (
    bill_number,
    student_id,
    total_fee,
    fee_due,
    fee_paid
  )
VALUES
  ('f001', 1, '50000.00', '10000.00', '40000.00'),
  ('f002', 2, '60000.00', '20000.00', '40000.00'),
  ('f003', 3, '20000.00', '5000.00', '15000.00'),
  ('f004', 4, '20000.00', '0.00', '20000.00'),
  ('f005', 5, '20000.00', '20000.00', '0.00'),
  ('f006', 6, '25000.00', '15000.00', '10000.00'),
  ('f007', 7, '70000.00', '10000.00', '60000.00'),
  ('f008', 8, '20000.00', '1500.00', '18500.00'),
  ('f009', 9, '25000.00', '20000.00', '5000.00'),
  ('f010', 10, '60000.00', '20000.00', '40000.00'),
  ('f011', 11, '60000.00', '10000.00', '50000.00'),
  ('f012', 12, '62000.00', '22000.00', '40000.00'),
  ('f013', 13, '70000.00', '10000.00', '60000.00'),
  ('f014', 14, '60000.00', '10000.00', '50000.00'),
  ('f015', 15, '40000.00', '20000.00', '20000.00'),
  ('f016', 16, '20000.50', '6000.50', '14000.00'),
  ('f017', 17, '30000.00', '10000.00', '20000.00'),
  ('f018', 18, '20000.50', '10000.50', '10000.00'),
  ('f019', 19, '30000.00', '12000.00', '18000.00'),
  ('f020', 20, '60000.50', '20000.00', '40000.50'),
```

```
('f021', 21, '70000.00', '10000.00', '60000.00'),
('f022', 22, '40000.00', '0.00', '40000.00'),
('f023', 23, '90000.00', '0.00', '90000.00'),
('f024', 24, '40000.00', '0.00', '40000.00'),
('f025', 25, '60000.00', '40000.00', '20000.00'),
('f026', 26, '70000.00', '40000.00', '30000.00'),
('f027', 27, '40000.00', '0.00', '40000.00'),
('f028', 28, '40000.00', '1000.00', '39000.00'),
('f029', 29, '50000.00', '5000.00', '45000.00'),
('f030', 30, '70000.00', '10000.00', '60000.00');
```

Result:

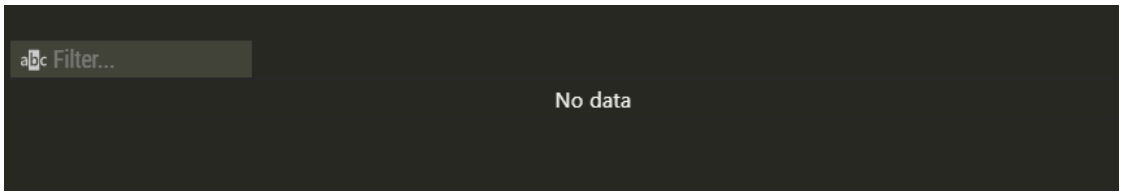


1. Creating admission table

Query:

```
CREATE TABLE admission (
  admission_id varchar(4) PRIMARY KEY,
  admission_date date,
  admission_fee decimal(8, 2),
  student_name varchar(50)
)
```

Result:



2. Describing admission table

Query:

```
desc admission;
```

Result:

Field	Type	Null	Key	Default	Extra
admission_id	varchar(4)	NO	PRI	NULL	
admission_date	date	YES		NULL	
admission_fee	decimal(8,2)	YES		NULL	
student_name	varchar(50)	YES		NULL	

3. Inserting values into admission table.

Query:

```
INSERT INTO admission (
    admission_id,
    admission_date,
```

```
    admission_fee,
    student_name
  )
VALUES
  (
    'AD1',
    '2076-01-02',
    '15000.00',
    'Roshan Parajuli'
  ),
  (
    'AD10',
    '2076-01-06',
    '20000.00',
    'Arjun Bhat'
  ),
  (
    'AD11',
    '2076-01-07',
    '25000.00',
    'Nikhil Upreti'
  ),
  (
    'AD12',
    '2076-01-10',
    '15000.00',
    'Bijay Dangal'
  ),
  (
    'AD13',
```

```
    '2076-01-12',
    '25000.00',
    'Abishek Shrestha'
  ),
  (
    'AD14',
    '2076-01-13',
    '25000.00',
    'Sandesh Khatiwada'
  ),
  (
    'AD15',
    '2076-01-13',
    '25000.00',
    'Saman Tamrakar'
  ),
  (
    'AD16',
    '2076-01-14',
    '25000.00',
    'Rohan Dahal'
  ),
  (
    'AD17',
    '2076-01-15',
    '25000.00',
    'Suman Khanal'
  ),
  (
    'AD18',
```

```
    '2076-01-15',
    '25000.00',
    'Saugat Acharya'
  ),
  (
    'AD19',
    '2076-01-20',
    '20000.00',
    'Rohit Parajuli'
  ),
  (
    'AD2',
    '2076-01-02',
    '15000.00',
    'Prakriti Dhakal'
  ),
  (
    'AD20',
    '2076-01-25',
    '25000.00',
    'Sumanta Dangal'
  ),
  (
    'AD21',
    '2076-01-25',
    '25000.00',
    'Urjit Bhattarai'
  ),
  (
    'AD22',
```

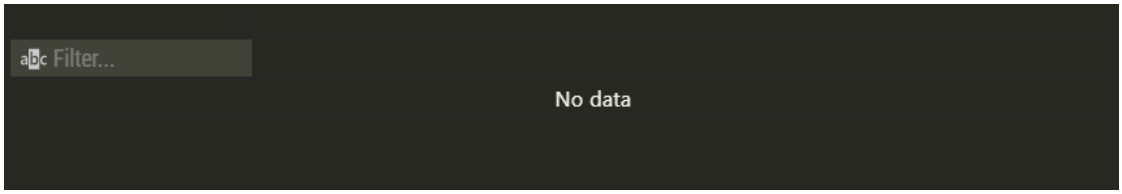
```
    '2076-01-26',
    '25000.00',
    'Sanam Dulal'
  ),
  (
    'AD23',
    '2076-01-25',
    '25000.00',
    'Pukar Dulal'
  ),
  (
    'AD24',
    '2076-01-25',
    '25000.00',
    'Ranbir Dulal'
  ),
  (
    'AD25',
    '2076-02-10',
    '15000.00',
    'Ishan Banjara'
  ),
  (
    'AD26',
    '2076-02-12',
    '25000.00',
    'Arun Dahal'
  ),
  (
    'AD27',
```

```
    '2076-02-12',
    '25000.00',
    'Sukhadev Acharya'
  ),
  (
    'AD28',
    '2076-02-12',
    '25000.00',
    'Suyog Baskota'
  ),
  (
    'AD29',
    '2077-01-01',
    '10000.00',
    'Shreejan Satyal'
  ),
  (
    'AD3',
    '2076-01-02',
    '15000.00',
    'Billu Prasad'
  ),
  (
    'AD30',
    '2077-01-02',
    '10000.00',
    'Arjun Parajuli'
  ),
  (
    'AD4',
```

```
    '2076-01-03',
    '20000.00',
    'Maya Kafle'
  ),
  (
    'AD5',
    '2076-01-03',
    '20000.00',
    'Rita Dangol'
  ),
  (
    'AD6',
    '2076-01-03',
    '20000.00',
    'Dristi Sigdel'
  ),
  (
    'AD7',
    '2076-01-03',
    '20000.00',
    'Sugandhi Chaudhary'
  ),
  (
    'AD8',
    '2076-01-04',
    '20000.00',
    'Saumya Bhandari'
  ),
  (
    'AD9',
```

```
    '2076-01-05',
    '20000.00',
    'Aayush Shrestha'
);
```

Result:

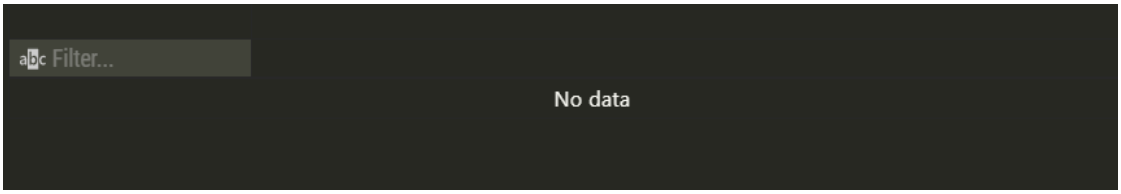


1. Creating result table

   Query:

```
CREATE TABLE result (
  student_id int(11),
  subject_id varchar(4),
  full_marks int(11),
  pass_marks int(11),
  marks_obtained decimal(4, 2),
  grade varchar(2),
  FOREIGN KEY (student_id) REFERENCES student(
  student_id),
  FOREIGN KEY (subject_id) REFERENCES subject(
  subject_id)
)
```

   Result:



2. Describing result table

   Query:

```
desc result;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
student_id	int(11)	NO	MUL	NULL	
subject_id	varchar(4)	NO	MUL	NULL	
full_marks	int(11)	YES		NULL	
pass_marks	int(11)	YES		NULL	
marks_obtained	decimal(4,2)	YES		NULL	
grade	varchar(2)	YES		NULL	

3. Inserting values into result table.

   Query:

```
INSERT INTO result (
    student_id,
    subject_id,
    full_marks,
    pass_marks,
    marks_obtained,
    grade
  )
VALUES
  (1, 'A10', 100, 40, '87.00', 'A'),
  (1, 'A2', 100, 40, '90.00', 'A+'),
  (1, 'A15', 100, 40, '75.00', 'B+'),
  (1, 'A5', 100, 40, '90.00', 'A+'),
  (1, 'A1', 100, 40, '75.00', 'B+'),
  (1, 'A4', 100, 40, '65.00', 'B'),
  (1, 'A6', 100, 40, '45.00', 'C'),
  (1, 'A8', 100, 40, '90.00', 'A+'),
  (1, 'A2', 100, 40, '75.00', 'B+'),
  (1, 'A11', 100, 40, '65.00', 'B'),
  (2, 'A11', 100, 40, '90.00', 'A+'),
  (2, 'A21', 100, 40, '79.00', 'B+'),
  (2, 'A13', 100, 40, '45.00', 'C'),
  (2, 'A14', 100, 40, '98.00', 'A+'),
  (2, 'A15', 100, 40, '79.00', 'B+'),
  (2, 'A16', 100, 40, '55.00', 'C+'),
  (2, 'A17', 100, 40, '45.00', 'C'),
```

```
(2, 'A7', 100, 40, '95.00', 'A+'),
(2, 'A9', 100, 40, '77.00', 'B+'),
(2, 'A10', 100, 40, '68.00', 'B'),
(3, 'A11', 100, 40, '75.00', 'B+'),
(3, 'A21', 100, 40, '89.00', 'A'),
(3, 'A13', 100, 40, '93.00', 'A+'),
(3, 'A14', 100, 40, '91.00', 'A+'),
(3, 'A15', 100, 40, '45.00', 'C'),
(3, 'A16', 100, 40, '55.00', 'C+'),
(3, 'A17', 100, 40, '65.00', 'B'),
(3, 'A18', 100, 40, '92.00', 'A+'),
(3, 'A19', 100, 40, '79.00', 'B+'),
(3, 'A20', 100, 40, '62.00', 'B'),
(4, 'A5', 100, 40, '65.00', 'B'),
(4, 'A7', 100, 40, '77.00', 'B+'),
(4, 'A9', 100, 40, '90.00', 'A+'),
(4, 'A11', 100, 40, '95.00', 'A+'),
(4, 'A13', 100, 40, '71.00', 'B+'),
(4, 'A15', 100, 40, '85.00', 'A'),
(4, 'A17', 100, 40, '55.00', 'C+'),
(4, 'A18', 100, 40, '91.00', 'A+'),
(4, 'A19', 100, 40, '77.00', 'B+'),
(4, 'A20', 100, 40, '67.00', 'B'),
(5, 'A1', 100, 40, '67.00', 'B'),
(5, 'A2', 100, 40, '75.00', 'B+'),
(5, 'A3', 100, 40, '92.00', 'A+'),
(5, 'A4', 100, 40, '88.00', 'A'),
(5, 'A5', 100, 40, '61.00', 'B'),
(5, 'A6', 100, 40, '88.00', 'A'),
(5, 'A7', 100, 40, '85.00', 'A'),
```

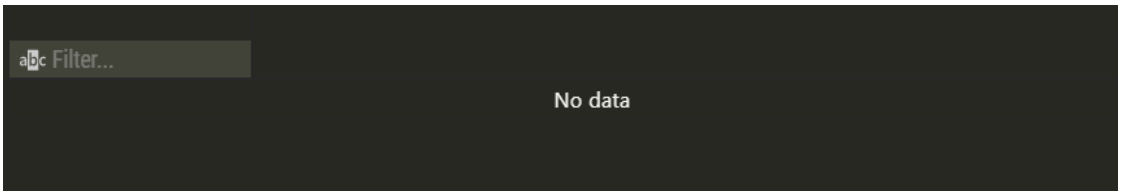
```
(5, 'A8', 100, 40, '91.00', 'A+'),
(5, 'A9', 100, 40, '97.00', 'A+'),
(5, 'A10', 100, 40, '92.00', 'A+'),
(6, 'A21', 100, 40, '95.00', 'A+'),
(6, 'A12', 100, 40, '65.00', 'B'),
(6, 'A23', 100, 40, '92.00', 'A+'),
(6, 'A24', 100, 40, '93.00', 'A+'),
(6, 'A25', 100, 40, '77.00', 'B+'),
(6, 'A26', 100, 40, '89.00', 'A'),
(6, 'A27', 100, 40, '56.00', 'C+'),
(6, 'A28', 100, 40, '93.00', 'A+'),
(6, 'A9', 100, 40, '75.00', 'B+'),
(6, 'A10', 100, 40, '61.00', 'C'),
(7, 'A21', 100, 40, '75.00', 'B+'),
(7, 'A2', 100, 40, '89.00', 'A'),
(7, 'A23', 100, 40, '93.00', 'A+'),
(7, 'A4', 100, 40, '91.00', 'A+'),
(7, 'A25', 100, 40, '45.00', 'C'),
(7, 'A6', 100, 40, '55.00', 'C+'),
(7, 'A27', 100, 40, '65.00', 'B'),
(7, 'A8', 100, 40, '92.00', 'A+'),
(7, 'A29', 100, 40, '79.00', 'B+'),
(7, 'A10', 100, 40, '62.00', 'B'),
(8, 'A1', 100, 40, '75.00', 'B+'),
(8, 'A12', 100, 40, '77.00', 'B+'),
(8, 'A23', 100, 40, '91.00', 'A+'),
(8, 'A4', 100, 40, '98.00', 'A+'),
(8, 'A5', 100, 40, '73.00', 'B+'),
(8, 'A6', 100, 40, '88.00', 'A'),
(8, 'A7', 100, 40, '61.00', 'B'),
```

```
(8, 'A8', 100, 40, '92.00', 'A+'),
(8, 'A9', 100, 40, '79.00', 'B+'),
(8, 'A10', 100, 40, '92.00', 'A+'),
(9, 'A1', 100, 40, '69.00', 'B'),
(9, 'A2', 100, 40, '77.00', 'B+'),
(9, 'A3', 100, 40, '94.00', 'A+'),
(9, 'A4', 100, 40, '92.00', 'A+'),
(9, 'A5', 100, 40, '79.00', 'B+'),
(9, 'A6', 100, 40, '80.00', 'A'),
(9, 'A7', 100, 40, '51.00', 'C+'),
(9, 'A8', 100, 40, '96.00', 'A+'),
(9, 'A9', 100, 40, '71.00', 'B+'),
(9, 'A10', 100, 40, '66.00', 'B'),
(10, 'A1', 100, 40, '91.00', 'A+'),
(10, 'A2', 100, 40, '71.00', 'B+'),
(10, 'A3', 100, 40, NULL, NULL),
(10, 'A4', 100, 40, '75.00', 'B+'),
(10, 'A5', 100, 40, '76.00', 'B+'),
(10, 'A6', 100, 40, '87.00', 'A'),
(10, 'A7', 100, 40, '58.00', 'C+'),
(10, 'A8', 100, 40, NULL, NULL),
(10, 'A9', 100, 40, '70.00', 'B+'),
(10, 'A10', 100, 40, '62.00', 'B'),
(11, 'A1', 100, 40, NULL, NULL),
(11, 'A2', 100, 40, '45.00', 'C'),
(11, 'A3', 100, 40, NULL, NULL),
(11, 'A4', 100, 40, NULL, NULL),
(11, 'A5', 100, 40, NULL, NULL),
(11, 'A6', 100, 40, '12.00', 'D'),
(11, 'A7', 100, 40, NULL, NULL),
```

```
(11, 'A8', 100, 40, NULL, NULL),
(11, 'A9', 100, 40, NULL, NULL),
(11, 'A10', 100, 40, '63.00', 'B'),
(12, 'A2', 100, 40, '18.00', 'D'),
(12, 'A3', 100, 40, '9.00', 'D'),
(12, 'A4', 100, 40, '8.00', 'D'),
(12, 'A5', 100, 40, '15.00', 'D'),
(12, 'A6', 100, 40, '55.00', 'C+'),
(12, 'A7', 100, 40, '5.00', 'D'),
(12, 'A8', 100, 40, '6.00', 'D'),
(12, 'A9', 100, 40, NULL, NULL),
(12, 'A10', 100, 40, NULL, NULL),
(13, 'A1', 100, 40, '91.00', 'A+'),
(13, 'A2', 100, 40, '72.00', 'B+'),
(13, 'A3', 100, 40, '43.00', 'C'),
(13, 'A4', 100, 40, '95.00', 'A+'),
(13, 'A5', 100, 40, '74.00', 'B+'),
(13, 'A6', 100, 40, '53.00', 'C+'),
(13, 'A7', 100, 40, '45.00', 'C'),
(13, 'A8', 100, 40, '97.00', 'A+'),
(13, 'A9', 100, 40, '78.00', 'B+'),
(13, 'A10', 100, 40, '68.00', 'B'),
(14, 'A1', 100, 40, '61.00', 'B'),
(14, 'A2', 100, 40, '73.00', 'B+'),
(14, 'A3', 100, 40, '94.00', 'A+'),
(14, 'A4', 100, 40, '92.00', 'A+'),
(14, 'A5', 100, 40, '72.00', 'B+'),
(14, 'A6', 100, 40, '83.00', 'A'),
(14, 'A7', 100, 40, '54.00', 'C+'),
(14, 'A8', 100, 40, '97.00', 'A+'),
```

```
(14, 'A9', 100, 40, '74.00', 'B+'),
(14, 'A10', 100, 40, '67.00', 'B'),
(15, 'A1', 100, 40, '69.00', 'B'),
(15, 'A2', 100, 40, '95.00', 'A+'),
(15, 'A3', 100, 40, '92.00', 'A+'),
(15, 'A4', 100, 40, '88.00', 'A'),
(15, 'A5', 100, 40, '61.00', 'B'),
(15, 'A6', 100, 40, '88.00', 'A'),
(15, 'A7', 100, 40, '85.00', 'A'),
(15, 'A8', 100, 40, '91.00', 'A+'),
(15, 'A9', 100, 40, '97.00', 'A+'),
(15, 'A10', 100, 40, '13.00', 'D');
```

Result:

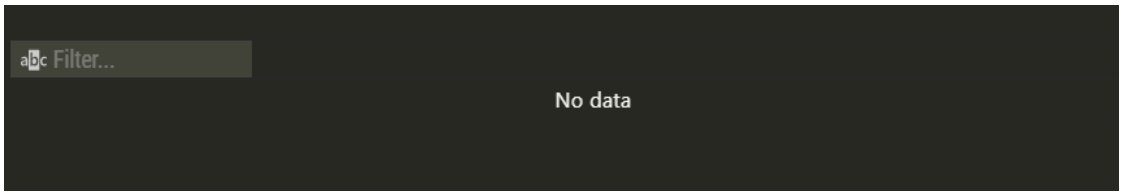


1. Creating transportation table

   Query:

```
CREATE TABLE transportation (
  vehicle_number varchar(20) PRIMARY KEY,
  vehicle_route varchar(50),
  vehicle_capacity int(11),
  transportation_mode varchar(20),
  transportation_cost decimal(6,2)
);
```

   Result:



2. Describing transportation table

   Query:

```
desc transportation;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
vehicle_number	varchar(20)	NO	PRI	NULL	
vehicle_route	varchar(50)	YES		NULL	
vehicle_capacity	int(11)	YES		NULL	
transportation_mode	varchar(20)	YES		NULL	
transportation_cost	decimal(6,2)	YES		NULL	

3. Inserting values into transportation table

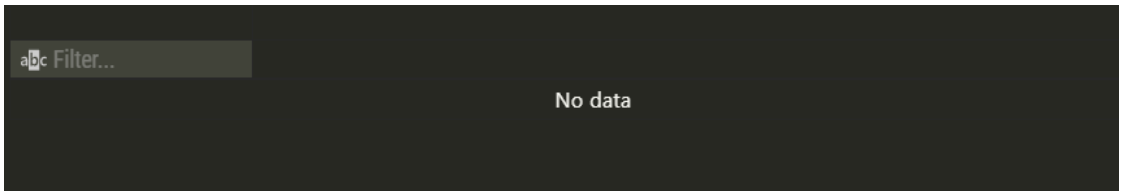
   Query:

```
INSERT INTO transportation (
    vehicle_number,
    vehicle_route,
    vehicle_capacity,
    transportation_mode,
    transportation_cost
  )
VALUES
  (
    'Ba 1 Cha 3456',
    'Chabahil - Sankhu',
    60,
    'Bus',
    '2000.00'
  ),
  (
    'Ba 1 Ja 6666',
    'Chabahil - Sanobharyang',
    30,
    'Van',
    '2000.00'
  ),
  (
    'Ba 1 Ka 3406',
    'Chabahil - Maharajgunj',
    20,
    'Van',
    '2000.00'
  ),
  (
```

```
    'Ba 1 Sa 1406',
    'Thimi - Bhaktapur',
    60,
    'Bus',
    '2500.00'
  ),
  (
    'Ba 2 Kha 4038',
    'Chabahil - Gothatar',
    80,
    'Bus',
    '2200.00'
  ),
  (
    'Ba 2 Sa 3406',
    'Gaushala - DharmaChowk',
    60,
    'Bus',
    '2400.00'
  ),
  (
    'Ba 3 Ha 3000',
    'Jorpati - Aryalgaun',
    30,
    'Van',
    '2200.00'
  ),
  (
    'Ba 8 Ja 2006',
    'Gaushala - Airport',
```

```
    20,
    'Van',
    '2500.00'
  ),
  (
    'Ba 8 Ka 2406',
    'Gaushala - Baneshwor',
    60,
    'Bus',
    '2000.00'
  ),
  (
    'Ba 8 Ka 6890',
    'Chabahil - NMC',
    60,
    'Bus',
    '2200.00'
  );
```

Result:

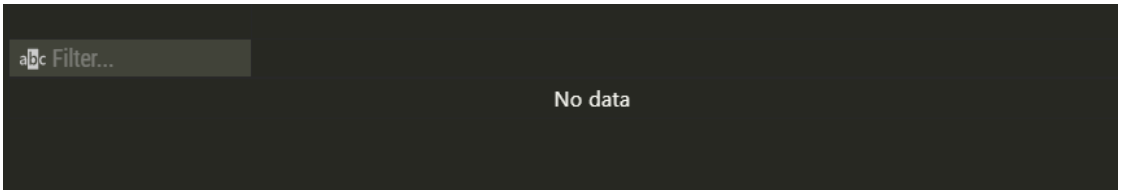


1. Creating student transportation table

   Query:

```
CREATE TABLE student_transport (
  student_id int(11) NOT NULL,
  vehicle_number varchar(20),
  FOREIGN KEY (student_id) REFERENCES
  student (student_id),
  FOREIGN KEY (vehicle_number) REFERENCES
  transportation (vehicle_number)
);
```

   Result:



2. Describing student transportation table

   Query:

```
desc student_transport;
```

   Result:

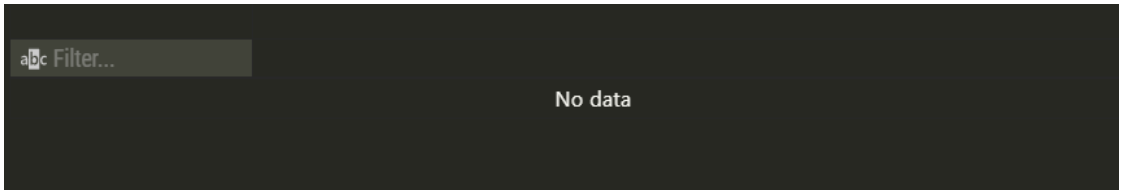
Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
vehicle_number	varchar(20)	NO	PRI	NULL	
vehicle_route	varchar(50)	YES		NULL	
vehicle_capacity	int(11)	YES		NULL	
transportation_mode	varchar(20)	YES		NULL	
transportation_cost	decimal(6,2)	YES		NULL	

3. Inserting values into student transportation table

   Query:

```
INSERT INTO student_transport (
  student_id,
  vehicle_number
  )
  VALUES
  (1, 'Ba 1 Cha 3456'),
  (2, 'Ba 1 Cha 3456'),
  (4, 'Ba 8 Ka 6890'),
  (5, 'Ba 3 Ha 3000'),
  (6, 'Ba 1 Ja 6666'),
  (7, 'Ba 1 Ka 3406'),
  (8, 'Ba 1 Sa 1406'),
  (9, 'Ba 2 Kha 4038'),
  (10, 'Ba 8 Ja 2006'),
  (11, 'Ba 3 Ha 3000'),
  (12, 'Ba 8 Ja 2006'),
  (13, 'Ba 8 Ka 6890'),
  (14, 'Ba 8 Ja 2006'),
  (15, 'Ba 8 Ja 2006');
```

Result:

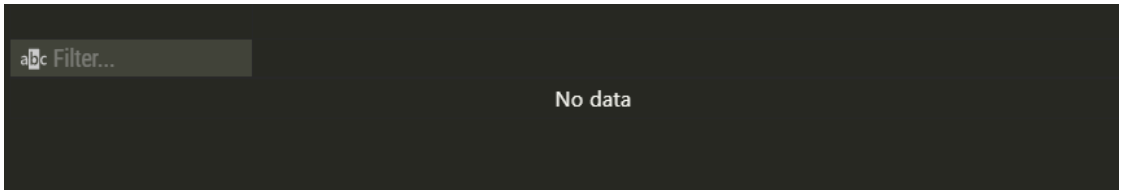


1. Creating food (menu) table

   Query:

```
CREATE TABLE food (
  food_name varchar(20),
  price int(11),
  PRIMARY KEY (food_name)
)
```

   Result:



2. Describing food table

   Query:

```
desc food;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
food_name	varchar(20)	NO	PRI	NULL	
price	int(11)	NO		NULL	

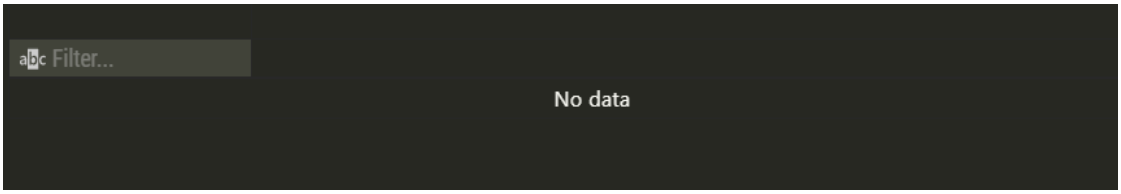
3. Inserting values into food table

   Query:

```
INSERT INTO food (food_name, price)
VALUES
  ('Black coffee', 25),
  ('Black tea', 15),
```

```
('Buff chowmein', 100),
('Buff momo', 100),
('Burger', 70),
('Cake', 60),
('Chana aalu', 30),
('Chicken chowmein', 120),
('Chicken momo', 120),
('Chowchow fry', 40),
('Chowmein', 70),
('Cold drinks', 35),
('Donuts', 15),
('French fries', 30),
('Fried Rice', 70),
('Lassi', 50),
('Milk coffee', 50),
('Milk tea', 30),
('Paneer Tikka', 80),
('Paratha', 120),
('Pasta', 60),
('Pizza', 80),
('Puri', 20),
('Roti', 30),
('Samosha', 40),
('Sandwitch', 40),
('Sel roti', 30),
('Tarkari', 20),
('Veg chowmein', 70),
('Veg momo', 70);
```

Result:

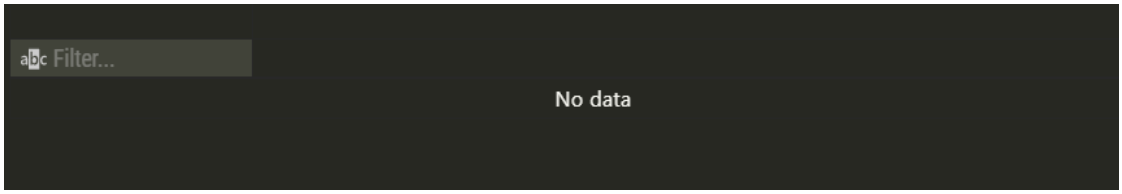

Filter...
No data

1. Creating canteen table

   Query:

```
CREATE TABLE canteen (
  student_id int(11),
  food_name varchar(20),
  FOREIGN KEY (student_id) REFERENCES student
  (student_id),
  FOREIGN KEY (food_name) REFERENCES food
  (food_name)
);
```

   Result:



2. Describing canteen table

   Query:

```
desc canteen;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
student_id	int(11)	NO	MUL	NULL	
food_name	varchar(20)	NO	MUL	NULL	

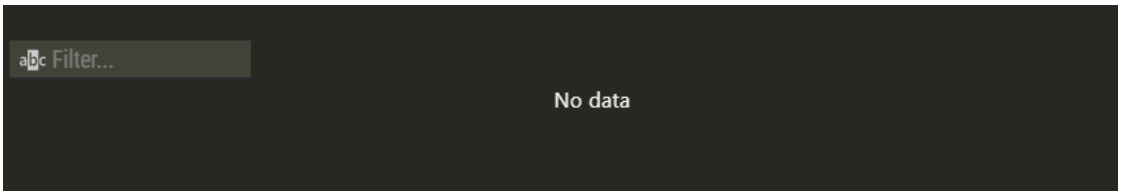
3. Inserting values into canteen table

   Query:

```
INSERT INTO canteen (student_id, food_name)
VALUES
  (8, 'Donuts'),
  (6, 'Chowchow fry'),
  (3, 'Tarkari'),
  (4, 'Lassi'),
  (19, 'Samosha'),
  (4, 'Veg momo'),
  (9, 'Roti'),
  (9, 'Tarkari'),
  (25, 'Pizza'),
  (23, 'Milk tea'),
  (18, 'Sel roti'),
  (29, 'Burger'),
  (5, 'Chana aalu'),
  (11, 'Pizza'),
  (17, 'Cake'),
  (4, 'Lassi'),
  (18, 'Donuts'),
  (16, 'Chowchow fry'),
  (30, 'Tarkari'),
  (7, 'Lassi'),
  (29, 'Samosha'),
  (7, 'Veg momo'),
  (19, 'Roti'),
  (8, 'Tarkari'),
  (5, 'Pizza'),
  (3, 'Milk tea'),
  (2, 'Sel roti'),
  (15, 'Burger'),
```

```
(16, 'Chana aalu'),
(17, 'Pizza'),
(7, 'Cake'),
(2, 'Lassi');
```

Result:

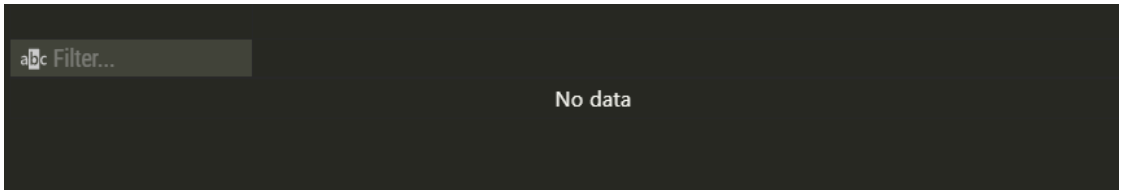


1. Creating teacher table

   Query:

```
CREATE TABLE teacher (
  staff_id varchar(10),
  subject_id varchar(4),
  FOREIGN KEY (staff_id) REFERENCES staff
  (staff_id),
  FOREIGN KEY (subject_id) REFERENCES subject
  (subject_id)
)
```

   Result:



2. Describing teacher table

   Query:

```
desc teacher;
```

   Result:

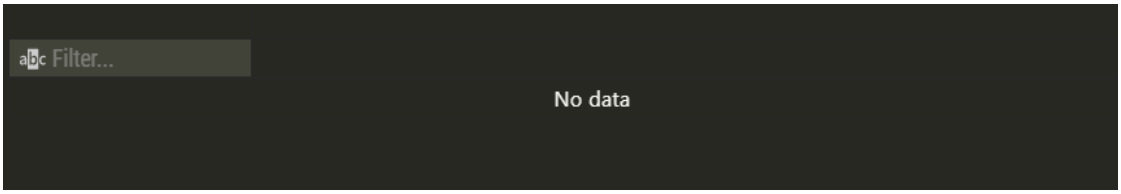
Field	Type	Null	Key	Default	Extra
staff_id	varchar(10)	YES	MUL	NULL	
subject_id	varchar(4)	YES	MUL	NULL	

3. Inserting values into teacher table

   Query:

```
INSERT INTO teacher (staff_id, subject_id)
VALUES
  ('S01', 'A9'),
  ('S02', 'A7'),
  ('S03', 'A2'),
  ('S05', 'A22'),
  ('S08', 'A10'),
  ('S10', 'A26'),
  ('S11', 'A1'),
  ('S12', 'A9'),
  ('S13', 'A17'),
  ('S14', 'A29'),
  ('S16', 'A27'),
  ('S17', 'A23'),
  ('S18', 'A6'),
  ('S19', 'A9'),
  ('S23', 'A1'),
  ('S26', 'A10'),
  ('S27', 'A11'),
  ('S28', 'A6'),
  ('S29', 'A5'),
  ('S30', 'A2');
```

Result:

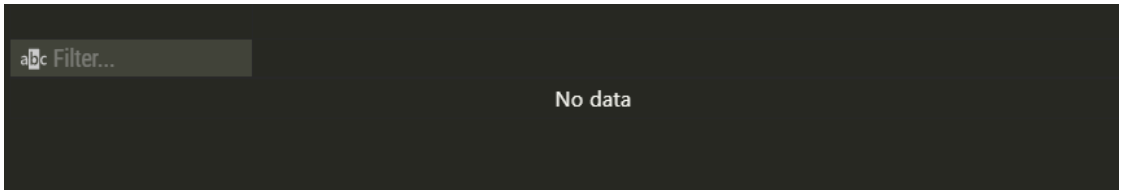


1. Creating student attendance table

   Query:

```
CREATE TABLE student_attendancee (
  student_id int,
  subject_id varchar(4),
  total_days int,
  present_days int,
  absent_days int AS (total_days - present_days),
  FOREIGN KEY (student_id) REFERENCES student
  (student_id),
  FOREIGN KEY (subject_id) REFERENCES subject
  (subject_id)
)
```

   Result:



2. Describing student attendance table

   Query:

```
desc student_attendance;
```

   Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
student_id	int(11)	NO	MUL	NULL	
subject_id	varchar(4)	NO	MUL	NULL	
total_days	int(11)	YES		NULL	
present_days	int(11)	YES		NULL	
absent_days	int(11)	YES		NULL	VIRTUAL GENERATED

3. Inserting values into student attendance table

Query:

```
INSERT INTO student_attendance (
    student_id,
    subject_id,
    total_days,
    present_days
  )
VALUES
  (1, 'A2', 90, 80),
  (1, 'A3', 95, 80),
  (1, 'A4', 90, 50),
  (1, 'A5', 90, 70),
  (1, 'A6', 90, 80),
  (1, 'A7', 90, 85),
  (1, 'A8', 90, 80),
  (1, 'A9', 90, 60),
  (1, 'A10', 90, 90),
  (2, 'A1', 90, 85),
  (2, 'A2', 90, 85),
  (2, 'A3', 95, 85),
  (2, 'A4', 90, 85),
  (2, 'A5', 90, 75),
  (2, 'A6', 90, 88),
  (2, 'A7', 90, 85),
  (2, 'A8', 90, 80),
  (2, 'A9', 90, 64),
  (2, 'A10', 90, 90),
  (4, 'A1', 90, 85),
```

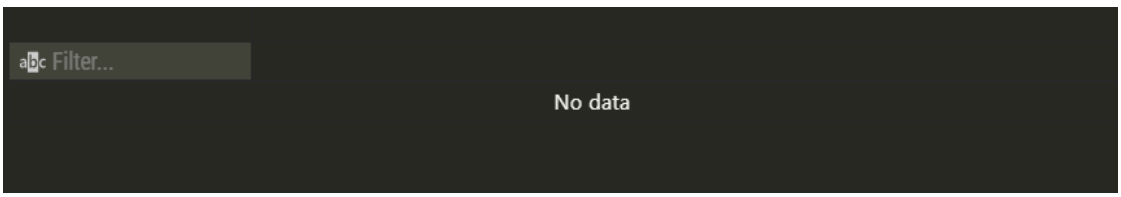
```
(4, 'A2', 90, 85),
(4, 'A3', 95, 85),
(4, 'A4', 90, 85),
(4, 'A5', 90, 75),
(4, 'A6', 90, 88),
(4, 'A7', 90, 85),
(4, 'A8', 90, 80),
(4, 'A9', 90, 64),
(4, 'A10', 90, 90),
(5, 'A1', 90, 85),
(5, 'A2', 90, 75),
(5, 'A3', 95, 75),
(5, 'A4', 90, 75),
(5, 'A5', 90, 75),
(5, 'A6', 90, 78),
(5, 'A7', 90, 75),
(5, 'A8', 90, 70),
(5, 'A9', 90, 64),
(5, 'A10', 90, 90),
(6, 'A2', 90, 79),
(6, 'A3', 95, 79),
(6, 'A4', 90, 76),
(6, 'A5', 90, 74),
(6, 'A6', 90, 77),
(6, 'A7', 90, 90),
(6, 'A8', 90, 87),
(6, 'A9', 90, 78),
(6, 'A10', 90, 90),
(7, 'A2', 90, 77),
(7, 'A3', 95, 76),
```

```
(7, 'A4', 90, 75),
(7, 'A5', 90, 75),
(7, 'A6', 90, 74),
(7, 'A7', 90, 76),
(7, 'A8', 90, 77),
(7, 'A9', 90, 68),
(7, 'A10', 90, 80),
(8, 'A2', 90, 77),
(8, 'A3', 95, 75),
(8, 'A4', 90, 73),
(8, 'A5', 90, 73),
(8, 'A6', 90, 77),
(8, 'A7', 90, 85),
(8, 'A8', 90, 80),
(8, 'A9', 90, 84),
(8, 'A10', 90, 85),
(9, 'A2', 90, 87),
(9, 'A3', 95, 77),
(9, 'A4', 90, 83),
(9, 'A5', 90, 78),
(9, 'A6', 90, 87),
(9, 'A7', 90, 88),
(9, 'A8', 90, 88),
(9, 'A9', 90, 88),
(9, 'A10', 90, 45),
(10, 'A2', 90, 77),
(10, 'A3', 95, 75),
(10, 'A4', 90, 76),
(10, 'A5', 90, 76),
(10, 'A6', 90, 77),
```

```
(10, 'A7', 90, 88),
(10, 'A8', 90, 86),
(10, 'A9', 90, 84),
(10, 'A10', 90, 87),
(11, 'A2', 90, 80),
(11, 'A3', 95, 80),
(11, 'A4', 90, 50),
(11, 'A5', 90, 70),
(11, 'A6', 90, 80),
(11, 'A7', 90, 85),
(11, 'A8', 90, 80),
(11, 'A9', 90, 60),
(11, 'A10', 90, 90),
(12, 'A1', 90, 85),
(12, 'A2', 90, 85),
(12, 'A3', 95, 85),
(12, 'A4', 90, 85),
(12, 'A5', 90, 75),
(12, 'A6', 90, 88),
(12, 'A7', 90, 85),
(12, 'A8', 90, 80),
(12, 'A9', 90, 64),
(12, 'A10', 90, 90),
(13, 'A1', 90, 86),
(13, 'A2', 90, 87),
(13, 'A3', 95, 85),
(13, 'A4', 90, 84),
(13, 'A5', 90, 75),
(13, 'A6', 90, 87),
(13, 'A7', 90, 84),
```

```
(13, 'A8', 90, 86),
(13, 'A9', 90, 66),
(13, 'A10', 90, 89),
(14, 'A1', 90, 85),
(14, 'A2', 90, 85),
(14, 'A3', 95, 85),
(14, 'A4', 90, 85),
(14, 'A5', 90, 75),
(14, 'A6', 90, 88),
(14, 'A7', 90, 85),
(14, 'A8', 90, 80),
(14, 'A9', 90, 64),
(14, 'A10', 90, 90),
(15, 'A1', 90, 85),
(15, 'A2', 90, 75),
(15, 'A3', 95, 75),
(15, 'A4', 90, 75),
(15, 'A5', 90, 75),
(15, 'A6', 90, 78),
(15, 'A7', 90, 75),
(15, 'A8', 90, 70),
(15, 'A9', 90, 64),
(15, 'A10', 90, 90);
```

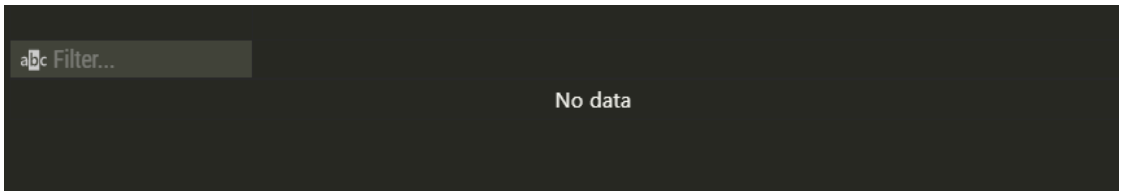
Result:



1. Creating staff attendance table

```
CREATE TABLE staff_attendance (
  staff_id varchar(10),
  total_days int,
  present_days int,
  absent_days int AS (total_days-present_days),
  FOREIGN KEY (staff_id) REFERENCES staff
  (staff_id)
);
```

Result:



1. Describing staff attendance table

Query:

```
desc staff_attendance;
```

Result:

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
staff_id	varchar(10)	YES	MUL	NULL	
total_days	int(11)	YES		NULL	
present_days	int(11)	YES		NULL	
absent_days	int(11)	YES		NULL	VIRTUAL GENERATED

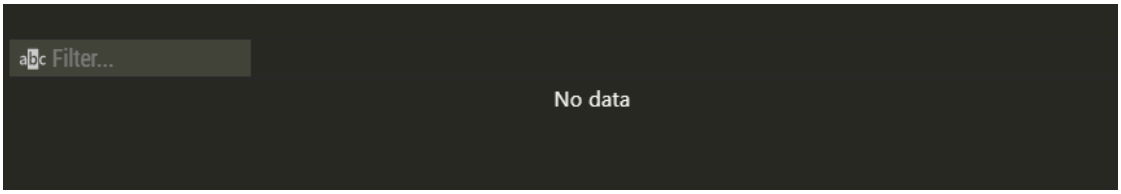
2. Inserting values into staff attendance table

Query:

```
INSERT INTO staff_attendance (
 staff_id,
 total_days,
 present_days
 )
VALUES
  ('S01', 90, 90),
  ('S02', 90, 85),
  ('S03', 90, 80),
  ('S04', 90, 85),
  ('S05', 90, 84),
  ('S06', 90, 79),
  ('S07', 90, 86),
  ('S08', 90, 80),
  ('S09', 90, 84),
  ('S10', 90, 80),
  ('S11', 95, 88),
  ('S12', 90, 85),
  ('S13', 90, 70),
  ('S14', 90, 70),
  ('S15', 90, 80),
  ('S16', 90, 85),
  ('S17', 90, 88),
  ('S18', 90, 89),
  ('S19', 90, 79),
  ('S20', 95, 80),
  ('S21', 90, 82),
  ('S22', 90, 85),
  ('S23', 95, 88),
  ('S24', 90, 75),
```

```
('S25', 90, 90),
('S26', 90, 88),
('S27', 90, 87),
('S28', 95, 89),
('S29', 90, 76),
('S30', 95, 93);
```

Result:

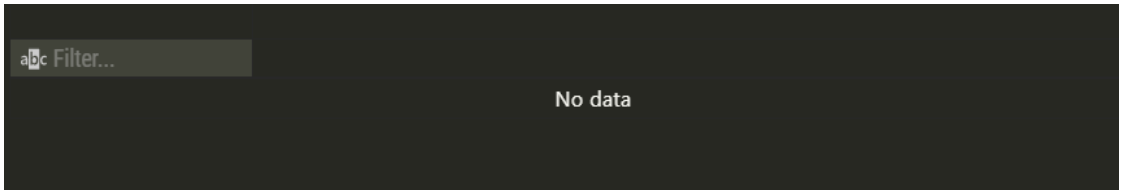


1. Creating exam table

   Query:

```
CREATE TABLE exam (
  exam_id int PRIMARY KEY AUTO_INCREMENT,
  subject_id varchar(4),
  exam_date date,
  exam_start_at time,
  exam_end_at time,
  class_id varchar(3),
  FOREIGN KEY (subject_id) REFERENCES subject
  (subject_id),
  FOREIGN KEY (class_id) REFERENCES class
  (class_id)
);
```

   Result:



2. Describing exam table

   Query:

```
desc exam;
```

   Result:

Field	Type	Null	Key	Default	Extra
exam_id	int(11)	NO	PRI	NULL	auto_increment
subject_id	varchar(4)	YES	MUL	NULL	
exam_date	date	YES		NULL	
exam_start_at	time	YES		NULL	
exam_end_at	time	YES		NULL	
class_id	varchar(3)	YES	MUL	NULL	

3. Inserting values into exam table

Query:

```
INSERT INTO exam (
    exam_id,
    subject_id,
    exam_date,
    exam_start_at,
    exam_end_at,
    class_id
  )
VALUES
  (
    1,
    'A1',
    '2077-03-14',
    '08:00:00',
    '10:00:00',
    'C01'
  ),
  (
    2,
    'A2',
    '2077-03-15',
    '10:00:00',
    '12:00:00',
    'C02'
  ),
  (
    3,
```

```
    'A3',
    '2077-03-17',
    '10:00:00',
    '12:00:00',
    'C07'
  ),
  (
    4,
    'A4',
    '2077-03-18',
    '08:00:00',
    '10:00:00',
    'C02'
  ),
  (
    5,
    'A5',
    '2077-03-19',
    '10:00:00',
    '12:00:00',
    'C10'
  ),
  (
    6,
    'A6',
    '2077-03-22',
    '06:00:00',
    '08:00:00',
    'C09'
  ),
```

```
  (
    7,
    'A7',
    '2077-03-23',
    '10:00:00',
    '12:00:00',
    'C06'
  ),
  (
    8,
    'A8',
    '2077-03-24',
    '10:00:00',
    '12:00:00',
    'C07'
  ),
  (
    9,
    'A9',
    '2077-03-25',
    '07:30:00',
    '09:30:00',
    'C02'
  ),
  (
    10,
    'A10',
    '2077-03-26',
    '10:00:00',
    '12:00:00',
```

```
    'C03'
  ),
  (
    11,
    'A11',
    '2077-03-27',
    '06:00:00',
    '08:00:00',
    'C03'
  ),
  (
    12,
    'A12',
    '2077-03-28',
    '06:00:00',
    '08:00:00',
    'C17'
  ),
  (
    13,
    'A13',
    '2077-03-29',
    '07:00:00',
    '09:00:00',
    'C18'
  ),
  (
    14,
    'A14',
    '2077-03-30',
```

```
    '06:00:00',
    '08:00:00',
    'C17'
  ),
  (
    15,
    'A15',
    '2077-04-01',
    '06:00:00',
    '08:00:00',
    'C17'
  ),
  (
    16,
    'A16',
    '2077-04-02',
    '10:00:00',
    '12:00:00',
    'C18'
  ),
  (
    17,
    'A17',
    '2077-04-03',
    '10:00:00',
    '12:00:00',
    'C19'
  ),
  (
    18,
```

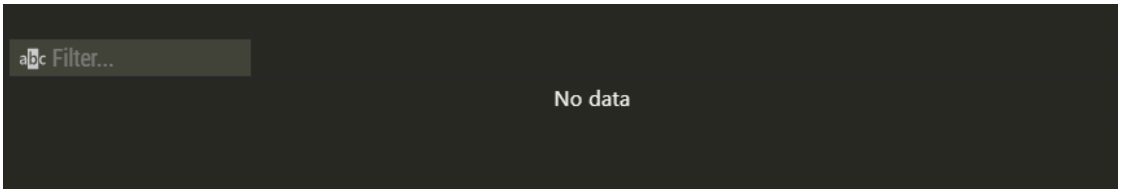
```
    'A18',
    '2077-04-05',
    '10:00:00',
    '12:00:00',
    'C03'
  ),
  (
    19,
    'A19',
    '2077-04-06',
    '10:00:00',
    '12:00:00',
    'C02'
  ),
  (
    20,
    'A20',
    '2077-04-07',
    '08:00:00',
    '10:00:00',
    'C15'
  ),
  (
    21,
    'A21',
    '2077-04-08',
    '06:00:00',
    '08:00:00',
    'C17'
  ),
```

```
(
  22,
  'A22',
  '2077-04-09',
  '07:00:00',
  '09:00:00',
  'C18'
),
(
  23,
  'A23',
  '2077-04-10',
  '06:00:00',
  '08:00:00',
  'C17'
),
(
  24,
  'A24',
  '2077-04-11',
  '06:00:00',
  '08:00:00',
  'C17'
),
(
  25,
  'A25',
  '2077-04-12',
  '10:00:00',
  '12:00:00',
```

```
    'C18'
  ),
  (
    26,
    'A26',
    '2077-04-13',
    '10:00:00',
    '12:00:00',
    'C19'
  ),
  (
    27,
    'A27',
    '2077-04-15',
    '10:00:00',
    '12:00:00',
    'C03'
  ),
  (
    28,
    'A28',
    '2077-04-16',
    '10:00:00',
    '12:00:00',
    'C12'
  ),
  (
    29,
    'A29',
    '2077-04-17',
```

```
    '08:00:00',
    '10:00:00',
    'C15'
  ),
  (
    30,
    'A30',
    '2077-04-18',
    '08:00:00',
    '10:00:00',
    'C15'
  );
```

Result:



7. SELECT statements using different functions

1.

```
SELECT
  student_id AS "ID",
  student_name AS "Name",
  student_address AS "Address",
  student_mobile AS "Mobile number",
  student_email AS "Email",
  student_dob AS "Date of birth"
FROM student
WHERE
  student_name LIKE 'A%'
ORDER BY
  student_id ASC;
```

Result:

ID	Name	Address	Mobile number	Email	Date of birth
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
9	Aayush Shrestha	Pashupati	9841573980	shresthaaaa@gmail.com	2057-01-01
10	Arjun Bhat	Bhaisepati	9840787878	arjun1@gmail.com	2059-12-12
13	Abishek Shrestha	Makalbari	9856049586	abishek.sh@yahoo.com	2052-05-04
26	Arun Dahal	Dakshin Dhoka	9841111780	arunman7@gmail.com	2058-03-08
30	Arjun Parajuli	Sankhu	9841003321	arjpj12@gmail.com	2057-09-25

2.

```
SELECT * FROM staff WHERE
  staff_job IN ('Cleaner','Warden','Chef');
```

Result:

staff_id	staff_name	staff_job	staff_dob	staff_email	staff_mobile
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
S04	Gajendra Kharel	Cleaner	2041-03-04	NULL	9851345678
S15	Rahul Dulal	Chef	2051-05-14	NULL	9840378712
S20	Aakash Shah	Chef	2057-09-13	NULL	NULL
S25	Roshan Bhardwaj	Warden	2047-05-12	bhardwajr1@gmail.com	9840567834

3.

```
SELECT * FROM staff WHERE
  staff_job = 'Cleaner'
  OR staff_job = 'Warden'
  OR staff_job = 'Chef';
```

Result:

staff_id	staff_name	staff_job	staff_dob	staff_email	staff_mobile
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
S04	Gajendra Kharel	Cleaner	2041-03-04	NULL	9851345678
S15	Rahul Dulal	Chef	2051-05-14	NULL	9840378712
S20	Aakash Shah	Chef	2057-09-13	NULL	NULL
S25	Roshan Bhardwaj	Warden	2047-05-12	bhardwajr1@gmail.com	9840567834

4.

```
SELECT student_id AS ID,
    UPPER(student_name) AS "Name"
FROM student
WHERE student_id NOT BETWEEN 5 AND 25;
```

Result:

ID	Name
Filter...	Filter...
1	ROSHAN PARAJULI
2	PRAKRITI DHAKAL
3	BILLU PRASAD
4	MAYA KAFLE
26	ARUN DAHAL
27	SUKHADEV ACHARYA
28	SUYOG BASKOTA
29	SHEEJAN SATYAL
30	ARJUN PARAJULI

5.

```
SELECT book_name,
    author
FROM book
WHERE author LIKE 'Chetan%'
LIMIT 3;
```

Result:

book_name	author
Filter...	Filter...
2 states	Chetan Bhagat
Revolution 2020	Chetan Bhagat
Night at Call Center	Chetan Bhagat

6.

```
SELECT *
FROM staff
WHERE staff_name LIKE '_an%';
```

Result:

staff_id	staff_name	staff_job	staff_dob	staff_email	staff_mobile
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
S02	Sangeet Baskota	Teacher	2039-03-04	baskotasan@gmail.com	9840378984
S05	Ganesh Bhattarai	Teacher	2051-09-14	ganeshbhatt21@gmail.com	9849134523
S07	Nanu Parajuli	Principal	2053-03-04	Nanubhattarai1@gmail.com	9840378790
S10	Manisha Bhattarai	Teacher	2047-06-02	manisha.sha@gmail.com	9840378456
S13	Pankaj Bihari	Teacher	2043-12-04	pakajbhaiya1@gmail.com	9840376740

7.

```
SELECT staff_name,
    staff_job
FROM staff
WHERE staff_email IS NULL
    AND staff_mobile IS NULL;
```

Result:

staff_name	staff_job
Filter...	Filter...
Aakash Shah	Chef

8.

```
SELECT AVG(price) AS "Average",
    MIN(price) AS "Minimum price",
    MAX(price) AS "Maximum price",
    COUNT(food_name) AS "Total items"
FROM food;
```

Result:

Average	Minimum price	Maximum price	Total items
Filter...	Filter...	Filter...	Filter...
56.3333	15	120	30

9.

```
SELECT student_id,
    food.food_name,
    price
FROM canteen
    INNER JOIN food
WHERE canteen.food_name = food.food_name
LIMIT 15;
;
```

Result:

student_id	food_name	price
Filter...	Filter...	Filter...
8	Donuts	15
6	Chowchow fry	40
3	Tarkari	20
4	Lassi	50
19	Samosha	40
4	Veg momo	70
9	Roti	30
9	Tarkari	20
25	Pizza	80
23	Milk tea	30
18	Sel roti	30
29	Burger	70
5	Chana aalu	30
11	Pizza	80
17	Cake	60

10.

```sql
SELECT DISTINCT student_name
FROM student
WHERE student_name NOT LIKE 'R%'
LIMIT 5;
```

Result:

student_name
Filter...
Prakriti Dhakal
Billu Prasad
Maya Kafle
Dristi Sigdel
Sugandhi Chaudhary

11.

```sql
SELECT staff_id
FROM staff
EXCEPT
SELECT staff_id
FROM teacher;
```

Result:

staff_id
Filter...
S04
S06
S07
S09
S15
S20
S21
S22
S24
S25

12.

```
SELECT *
FROM staff
WHERE staff_id IN (
        SELECT staff_id
        FROM staff
        EXCEPT
        SELECT staff_id
        FROM teacher
    );
```

Result:

staff_id	staff_name	staff_job	staff_dob	staff_email	staff_mobile
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
S04	Gajendra Kharel	Cleaner	2041-03-04	NULL	9851345678
S06	Jiban Pokhrel	Accountant	2055-08-04	jiban.pokhrel@gmail.com	9849123456
S07	Nanu Parajuli	Principal	2053-03-04	Nanubhattarai1@gmail.com	9840378790
S09	Laxman Adhikari	Watchman	2056-08-04	laxman.ad12@gmail.com	9840678799
S15	Rahul Dulal	Chef	2051-05-14	NULL	9840378712
S20	Aakash Shah	Chef	2057-09-13	NULL	NULL
S21	Minakshat Sharma	Driver	2047-09-13	minak@gmail.com	9840378765
S22	Ram Kharel	Driver	2047-08-13	ramkj12@gmail.com	9840378666
S24	Shaleshma Shah	Cleaning Lady	2047-09-13	singhshal1@gmail.com	9841378095
S25	Roshan Bhardwaj	Warden	2047-05-12	bhardwajr1@gmail.com	9840567834

13.

```
SELECT concat(book_id, " ", book_name)
FROM book
WHERE book_id IN (
        SELECT book_id
        FROM book
        WHERE book_name LIKE "%A%" OR "a%"
    );
```

Result:

concat(book_id,...
Filter...
1 2 states
4 Clear City
6 Circle of Reason
7 Death of City
8 Gravity
9 A Secular Agenda
10 Algebra
11 Bhagwat Geeta
12 Magic Seeds
14 Against the Day
15 Saya
19 Night at Call Center
20 Half Girlfriend

14.

```
SELECT *
FROM staff
WHERE staff_job ="Teacher" && staff_name LIKE 'aa%';
```

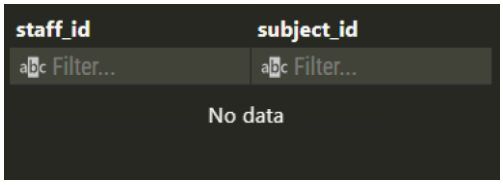
Result:

staff_id	staff_name	staff_job	staff_dob	staff_email	staff_mobile
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
No data					

15.

```
SELECT *
FROM teacher
WHERE EXISTS(
        SELECT staff_id
        FROM staff
        WHERE staff_job = "Teacher" && staff_name
        LIKE 'aa%'
    );
```

Result:



16.

```
SELECT *
FROM teacher
WHERE NOT EXISTS(
        SELECT staff_id
        FROM staff
        WHERE staff_job = "Cleaner" && staff_name
        LIKE 'aa%'
    )
LIMIT 5;
```

Result:

staff_id	subject_id
abc Filter...	abc Filter...
S01	A9
S02	A7
S03	A2
S05	A22
S08	A10

17.

```
SELECT *
FROM library
WHERE EXISTS (
        SELECT *
        FROM student
        WHERE student_id = 100
    );
```

Result:

student_id	borrowed_date	return_date	book_id
abc Filter...	abc Filter...	abc Filter...	abc Filter...
No data			

18.

```
SELECT DISTINCT staff_job,
    count(staff_job) AS "No. of staff"
FROM staff
GROUP BY staff_job
ORDER BY count(staff_job) DESC;
```

Result:

staff_job	No. of staff
Filter...	Filter...
Teacher	20
Driver	2
Chef	2
Cleaner	1
Cleaning Lady	1
Accountant	1
Warden	1
Principal	1
Watchman	1

19.

```
SELECT student_name
FROM student
WHERE student_id IN (
        SELECT DISTINCT student_id
        FROM library
        WHERE return_date BETWEEN '2077-09-01'
        AND '2077-12-12'
    );
```

Result:

student_name
Filter...
Rohit Parajuli
Sumanta Dangal
Sanam Dulal
Ishan Banjara
Sheejan Satyal
Arjun Parajuli

20.

```
(SELECT student_name
FROM student LIMIT 10)
UNION
(SELECT staff_name
FROM staff LIMIT 10);
```

Result:

student_id	student_name
Filter...	Filter...
1	Roshan Parajuli
2	Prakriti Dhakal
3	Billu Prasad
4	Maya Kafle
5	Rita Dangol
6	Dristi Sigdel
7	Sugandhi Chaudhary
8	Saumya Bhandari
9	Aayush Shrestha
10	Arjun Bhat
S01	Sudan Bhakta Acharya
S02	Sangeet Baskota
S03	Roshan Pyakurel
S04	Gajendra Kharel
S05	Ganesh Bhattarai
S06	Jiban Pokhrel
S07	Nanu Parajuli
S08	Ramesh Lal Yadav
S09	Laxman Adhikari
S10	Manisha Bhattarai

21.

```
SELECT vehicle_number
FROM student_transport
UNION ALL
SELECT vehicle_number
FROM transportation;
```

Result:

vehicle_number
Filter...
Ba 1 Cha 3456
Ba 1 Cha 3456
Ba 1 Ja 6666
Ba 1 Ka 3406
Ba 1 Sa 1406
Ba 2 Kha 4038
Ba 3 Ha 3000
Ba 3 Ha 3000
Ba 8 Ja 2006
Ba 8 Ja 2006
Ba 8 Ja 2006
Ba 8 Ja 2006
Ba 8 Ka 6890
Ba 8 Ka 6890
Ba 1 Cha 3456
Ba 1 Ja 6666
Ba 1 Ka 3406
Ba 1 Sa 1406
Ba 2 Kha 4038
Ba 2 Sa 3406
Ba 3 Ha 3000
Ba 8 Ja 2006
Ba 8 Ka 2406
Ba 8 Ka 6890

22.

```
SELECT subject_id
FROM teacher
INTERSECT
SELECT subject_id
FROM subject;
```

Result:

subject_id
Filter...
A1
A10
A11
A17
A2
A22
A23
A26
A27
A29
A5
A6
A7
A9

23.

```sql
SELECT *
FROM exam
WHERE exam_start_at LIKE '07:__:__'
    AND class_id = "C01"
    OR (
        class_id = "C07"
        AND subject_id NOT LIKE 'A_'
    );
```

Result:

exam_id	subject_id	exam_date	exam_start_at	exam_end_at	class_id
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1	A1	2077-03-14	07:30:00	09:30:00	C01

24.

```
SELECT staff_name
FROM staff
WHERE staff_name LIKE 'R_S_A%'
    AND staff_email LIKE '%@gmail.com';
```

Result:

staff_name
Filter...
Roshan Pyakurel
Roshan Bhardwaj

25.

```
SELECT student_name
FROM student
ORDER BY field (
        student_name,
        'Roshan Parajuli',
        'Prakriti Dhakal'
    ) DESC
LIMIT 5;
```

Result:

student_name
Filter...
Prakriti Dhakal
Roshan Parajuli
Yogeshwor Amatya
Maya Kafle
Rita Dangol

8. SELECT statement using Sub Query

1.

```
SELECT student_name
FROM student
WHERE student_id IN (
        SELECT student_id
        FROM student_attendance
        GROUP BY student_id
        HAVING SUM(present_days) > 700
    );
```

Result:

student_name
Filter...
Prakriti Dhakal
Maya Kafle
Rita Dangol
Dristi Sigdel
Saumya Bhandari
Aayush Shrestha
Arjun Bhat
Bijay Dangal
Abishek Shrestha
Sandesh Khatiwada
Saman Tamrakar

2.

```
SELECT food_name,
    count(food_name) AS "Sold Quantity"
FROM canteen
WHERE food_name IN (
        SELECT food_name
```

```
        FROM canteen
    )
GROUP BY food_name;
```

Result:

food_name	Sold Quantity
Filter...	Filter...
Burger	2
Cake	2
Chana aalu	2
Chowchow fry	2
Donuts	2
Lassi	4
Milk tea	2
Pizza	4
Roti	2
Samosha	2
Sel roti	2
Tarkari	4
Veg momo	2

3.

```
SELECT *
FROM subject
WHERE subject_id NOT IN (
        SELECT subject_id
        FROM teacher
    )
    AND subject_name LIKE '%math%';
```

Result:

subject_id	subject_name	subject_description
Filter...	Filter...	Filter...
A24	Advanced Maths	Pre Calculus and Algebra
A3	Mathematics	Linear Algebra and Logic

4.

```
SELECT student_name,
    admission_date
FROM admission
WHERE student_name IN (
        SELECT student_name
        FROM student
        WHERE student_id BETWEEN 1 AND 5
            OR student_id BETWEEN 10 AND 15
    );
```

Result:

student_name	admission_date
Filter...	Filter...
Roshan Parajuli	2076-01-02
Arjun Bhat	2076-01-06
Nikhil Upreti	2076-01-07
Bijay Dangal	2076-01-10
Abishek Shrestha	2076-01-12
Sandesh Khatiwada	2076-01-13
Saman Tamrakar	2076-01-13
Prakriti Dhakal	2076-01-02
Billu Prasad	2076-01-02
Maya Kafle	2076-01-03
Rita Dangol	2076-01-03

5.

```
SELECT vehicle_route,
        transportation_mode
FROM transportation
WHERE vehicle_number IN
    (SELECT vehicle_number
    FROM student_transport
```

```
    WHERE student_id = 1
            OR student_id = 7 );
```

Result:

vehicle_route	transportation_mode
abc Filter...	abc Filter...
Chabahil - Sankhu	Bus
Chabahil - Maharajgunj	Van

9. SELECT statement using Count and Group Functions

1.

```
SELECT admission_date,
    COUNT(admission_date) AS "Number of admissions"
FROM admission
GROUP BY admission_date
HAVING COUNT(admission_date) > 1;
```

Result:

admission_date	Number of admissions
Filter...	Filter...
2076-01-02	3
2076-01-03	4
2076-01-13	2
2076-01-15	2
2076-01-25	4
2076-02-12	3

2.

```
SELECT student_id,
    SUM(marks_obtained) AS "Total Marks",
    (SUM(marks_obtained) / SUM(full_marks)) * 100
    AS percentage
FROM result
GROUP BY student_id;
```

Result:

student_id	Total Marks	percentage
Filter...	Filter...	Filter...
1	757.00	75.700000
2	731.00	73.100000
3	746.00	74.600000
4	773.00	77.300000
5	836.00	83.600000
6	796.00	79.600000
7	746.00	74.600000
8	826.00	82.600000
9	775.00	77.500000
10	590.00	59.000000
11	120.00	12.000000
12	116.00	12.888889
13	716.00	71.600000
14	767.00	76.700000
15	779.00	77.900000

3.

```
SELECT book_id,
    COUNT(book_id) AS "Total Quantity"
FROM library
GROUP BY book_id;
```

Result:

book_id	Total Quantity
Filter...	Filter...
1	9
2	8
3	2
4	3
5	2

10. SELECT statement using Different Joins

1.

```
SELECT student.student_id,
    student_name,
    SUM AS "Total price",
    COUNT AS "Total items"
FROM student
    INNER JOIN (
        SELECT student_id,
            SUM(price) AS SUM,
            COUNT(price) AS COUNT
        FROM canteen
            INNER JOIN food ON canteen.food_name =
            food.food_name
        GROUP BY student_id
    ) AS x ON x.student_id = student.student_id;
```

Result:

student_id	student_name	Total price	Total items
Filter...	Filter...	Filter...	Filter...
2	Prakriti Dhakal	80	2
3	Billu Prasad	50	2
4	Maya Kafle	170	3
5	Rita Dangol	110	2
6	Dristi Sigdel	40	1
7	Sugandhi Chaudhary	180	3
8	Saumya Bhandari	35	2
9	Aayush Shrestha	50	2
11	Nikhil Upreti	80	1
15	Saman Tamrakar	70	1
16	Rohan Dahal	70	2
17	Suman Khanal	140	2
18	Saugat Acharya	45	2
19	Rohit Parajuli	70	2
23	Pukar Dulal	30	1
25	Ishan Banjara	80	1
29	Sheejan Satyal	110	2
30	Arjun Parajuli	20	1

2.

```
SELECT subject.subject_id,
    staff_id,
    subject_name,
    subject_description
FROM subject
    LEFT JOIN teacher ON subject.subject_id =
    teacher.subject_id
WHERE subject.subject_id LIKE 'A1_'
ORDER BY subject.subject_id DESC;
```

Result:

subject_id	staff_id	subject_name	subject_description
Filter...	Filter...	Filter...	Filter...
A19	NULL	Political Science	History and present condition of society
A18	NULL	Personal Development	The development of one's character
A17	S13	EPH	Health, Population and Environment
A16	NULL	Nepali Vyakaran	Basic Nepali grammar
A15	NULL	English Grammar	Basic English grammar
A14	NULL	History	Brief history of Nepal
A13	NULL	Accounts	Basic Principles of Accounting
A12	NULL	GK	General Knowledge of important dates and eve...
A11	S27	Moral Science	Valuable lessons through stories
A10	S08	Nepali	Nepali stories and Byakaran
A10	S26	Nepali	Nepali stories and Byakaran

3.

```
SELECT student_name,
    student_address,
    student_transport.student_id,
    student_transport.vehicle_number
FROM student_transport
    RIGHT JOIN student ON student_transport.student
_id = student.student_id
```

```
WHERE student_address IN ('Narayantar', 'Aryalgaun'
, 'Dakshin Dhoka');
```

Result:

student_name	student_address	student_id	vehicle_number
Filter...	Filter...	Filter...	Filter...
Roshan Parajuli	Dakshin Dhoka	1	Ba 1 Cha 3456
Maya Kafle	Narayantar	4	Ba 8 Ka 6890
Bijay Dangal	Dakshin Dhoka	12	Ba 8 Ja 2006
Suman Khanal	Narayantar	NULL	NULL
Rohit Parajuli	Narayantar	NULL	NULL
Sumanta Dangal	Aryalgaun	NULL	NULL
Sanam Dulal	Narayantar	NULL	NULL
Pukar Dulal	Narayantar	NULL	NULL
Ranbir Dulal	Narayantar	NULL	NULL
Ishan Banjara	Narayantar	NULL	NULL
Arun Dahal	Dakshin Dhoka	NULL	NULL
Suyog Baskota	Narayantar	NULL	NULL
Sheejan Satyal	Narayantar	NULL	NULL

4.

```
SELECT library.book_id,
    library.student_id,
    book_name,
    author,
    borrowed_date,
    return_date
FROM book
    CROSS JOIN library ON book.book_id = library.
    book_id
ORDER BY borrowed_date ASC;
```

Result:

book_id	student_id	book_name	author	borrowed_date	return_date
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1	1	2 states	Chetan Bhagat	2077-01-05	2077-01-10
1	2	2 states	Chetan Bhagat	2077-01-05	2077-01-10
1	4	2 states	Chetan Bhagat	2077-01-06	2077-01-11
1	5	2 states	Chetan Bhagat	2077-01-07	2077-01-12
1	3	2 states	Chetan Bhagat	2077-01-07	2077-01-17
3	6	Development	Amartya Sen	2077-02-15	2077-02-20
3	7	Development	Amartya Sen	2077-03-05	2077-03-10
1	8	2 states	Chetan Bhagat	2077-03-06	2077-03-16
4	11	Clear City	Anita Desai	2077-04-06	2077-04-11
4	9	Clear City	Anita Desai	2077-04-06	2077-04-11
4	10	Clear City	Anita Desai	2077-04-06	2077-04-16
1	12	2 states	Chetan Bhagat	2077-05-06	2077-05-11
1	13	2 states	Chetan Bhagat	2077-06-08	2077-06-14
1	14	2 states	Chetan Bhagat	2077-08-06	2077-08-11
5	15	Other Musketeers	Alexander Dumas	2077-08-06	2077-08-16
2	17	The Three Musketeers	Alexander Dumas	2077-08-18	2077-08-28
2	16	The Three Musketeers	Alexander Dumas	2077-08-18	2077-08-28
2	18	The Three Musketeers	Alexander Dumas	2077-08-19	2077-08-29
2	19	The Three Musketeers	Alexander Dumas	2077-09-01	2077-09-11
2	20	The Three Musketeers	Alexander Dumas	2077-09-01	2077-09-11
2	22	The Three Musketeers	Alexander Dumas	2077-09-02	2077-09-12
2	25	The Three Musketeers	Alexander Dumas	2077-09-03	2077-09-13
2	29	The Three Musketeers	Alexander Dumas	2077-09-03	2077-09-13
5	30	Other Musketeers	Alexander Dumas	2077-09-07	2077-09-17

5.

```sql
SELECT staff.staff_id,
    staff.staff_name,
    total_days,
    present_days,
    absent_days
FROM staff_attendance
    INNER JOIN staff
WHERE staff_attendance.staff_id = staff.staff_id
    AND staff_job = "Teacher"
    AND absent_days<10;
```

Result:

staff_id	staff_name	total_days	present_days	absent_days
Filter...	Filter...	Filter...	Filter...	Filter...
S01	Sudan Bhakta Acharya	90	90	0
S02	Sangeet Baskota	90	85	5
S05	Ganesh Bhattarai	90	84	6
S11	Samyog Brahmachari	95	88	7
S12	Ramesh Maan	90	85	5
S16	Sashwat Bhandari	90	85	5
S17	Abninav Bahadur	90	88	2
S18	Ram Prasad Yadav	90	89	1
S23	Abishek Upmanyu	95	88	7
S26	Deepa Upadhyaye	90	88	2
S27	Junu Hada	90	87	3
S28	Arun Basyal	95	89	6
S30	Deepa Kuikel	95	93	2

6.

```
SELECT student_name,
    x.vehicle_number,
    vehicle_route,
    transportation_cost
FROM student
    INNER JOIN (
        SELECT student_id,
            vehicle_number,
            vehicle_route,
            transportation_cost
        FROM student_transport
            INNER JOIN transportation USING (
            vehicle_number)
    ) AS x ON x.student_id = student.student_id;
```

Result:

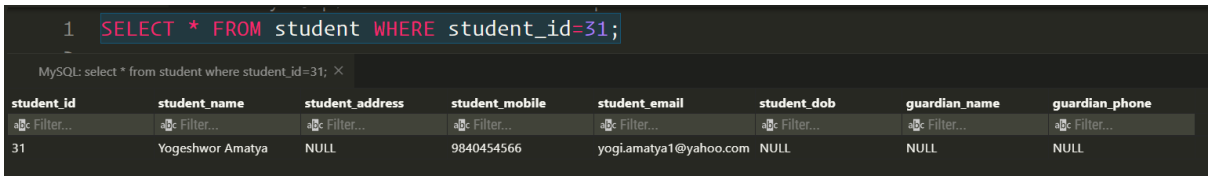
student_name	vehicle_number	vehicle_route	transportation_cost
Filter...	Filter...	Filter...	Filter...
Roshan Parajuli	Ba 1 Cha 3456	Chabahil - Sankhu	2000.00
Prakriti Dhakal	Ba 1 Cha 3456	Chabahil - Sankhu	2000.00
Saumya Bhandari	Ba 1 Cha 3456	Chabahil - Sankhu	2000.00
Arjun Bhat	Ba 1 Cha 3456	Chabahil - Sankhu	2000.00
Bijay Dangal	Ba 1 Cha 3456	Chabahil - Sankhu	2000.00
Sandesh Khatiwada	Ba 1 Cha 3456	Chabahil - Sankhu	2000.00
Saman Tamrakar	Ba 1 Cha 3456	Chabahil - Sankhu	2000.00
Dristi Sigdel	Ba 1 Ja 6666	Chabahil - Sanobharya...	2000.00
Sugandhi Chaudhary	Ba 1 Ka 3406	Chabahil - Maharajgunj	2000.00
Aayush Shrestha	Ba 2 Kha 4038	Chabahil - Gothatar	2200.00
Rita Dangol	Ba 3 Ha 3000	Jorpati - Aryalgaun	2200.00
Nikhil Upreti	Ba 3 Ha 3000	Jorpati - Aryalgaun	2200.00
Maya Kafle	Ba 8 Ka 6890	Chabahil - NMC	2200.00
Abishek Shrestha	Ba 8 Ka 6890	Chabahil - NMC	2200.00

11. Insert Statement

1.

```
INSERT INTO student (
        student_id,
        student_name,
        student_address,
        student_email,
        student_mobile
    ) VALUES (
        31,
        "Yogeshwor Amatya",
        DEFAULT,
        "yogi.amatya1@yahoo.com",
        "9840454566"
    );
```

Result:



2. Creating a temporary table “book_dup” to have its values duplicated from the original book table.

```
CREATE TABLE book_dup(
    book_id INT,
    book_name VARCHAR(20),
    author VARCHAR(30),
    edition VARCHAR(5),
    quantity INT
```

```
);
INSERT INTO book_dup (quantity, book_id, book_name)
VALUES(5, 21, "A random book");
```

```
INSERT INTO book_dup SELECT * FROM book LIMIT 5;
```

Result:

```
SELECT * FROM book_dup;
```

book_id	book_name	author	edition	quantity
Filter...	Filter...	Filter...	Filter...	Filter...
21	A random book	NULL	NULL	5
1	2 states	Chetan Bhagat	2nd	20
2	The Three Musketeers	Alexander Dumas	1st	10
3	Development	Amartya Sen	1st	10
4	Clear City	Anita Desai	2nd	2
5	Other Musketeers	Alexander Dumas	1st	10

3.

```
INSERT INTO book_dup (book_id, book_name, author)
SELECT book_id,
    book_name,
    author
FROM book
WHERE book_id > 5
    AND book_id < 8;
```

Result:

book_id	book_name	author	edition	quantity
Filter...	Filter...	Filter...	Filter...	Filter...
21	A random book	NULL	NULL	5
1	2 states	Chetan Bhagat	2nd	20
2	The Three Musketeers	Alexander Dumas	1st	10
3	Development	Amartya Sen	1st	10
4	Clear City	Anita Desai	2nd	2
5	Other Musketeers	Alexander Dumas	1st	10

4.

```
INSERT INTO book_dup (book_id, book_name, author)
SELECT book_id,
    book_name,
    author
FROM book
WHERE book_id = 9;
```

Result:

book_id	book_name	author	edition	quantity
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
21	A random book	NULL	NULL	5
1	2 states	Chetan Bhagat	2nd	20
2	The Three Musketeers	Alexander Dumas	1st	10
3	Development	Amartya Sen	1st	10
4	Clear City	Anita Desai	2nd	2
5	Other Musketeers	Alexander Dumas	1st	10
6	Circle of Reason	Amitav Ghose	NULL	NULL
7	Death of City	Amrita Pritam	NULL	NULL
9	A Secular Agenda	Arun Shourie	NULL	NULL

5.

```
INSERT INTO staff (
        staff_job,
        staff_email,
        staff_mobile,
        staff_name,
        staff_id
    )
VALUES(
    "Teacher",
        "pabitra.rai12@gmail.com",
        "9845347684",
```

```
        "Pabitra Rai",
        "S31"
)
```

Result:

```
1  SELECT * FROM staff WHERE staff_id='S31';
```

MySQL: SELECT * FROM staff WHERE staff_id='S31';

staff_id	staff_name	staff_job	staff_dob	staff_email	staff_mobile
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
S31	Pabitra Rai	Teacher	NULL	pabitra.rai12@gmail.com	9845347684

12. Update Statement

1.

book_dup table before updating:

book_id	book_name	author	edition	quantity
Filter...	Filter...	Filter...	Filter...	Filter...
21	A random book	NULL	NULL	5
1	2 states	Chetan Bhagat	2nd	20
2	The Three Musketeers	Alexander Dumas	1st	10
3	Development	Amartya Sen	1st	10
4	Clear City	Anita Desai	2nd	2
5	Other Musketeers	Alexander Dumas	1st	10
6	Circle of Reason	Amitav Ghose	NULL	NULL
7	Death of City	Amrita Pritam	NULL	NULL
9	A Secular Agenda	Arun Shourie	NULL	NULL

```
UPDATE book_dup SET quantity=10,edition='1st';
```

book_dup table after updating omitting the where clause:

book_id	book_name	author	edition	quantity
Filter...	Filter...	Filter...	Filter...	Filter...
21	A random book	NULL	1st	10
1	2 states	Chetan Bhagat	1st	10
2	The Three Musketeers	Alexander Dumas	1st	10
3	Development	Amartya Sen	1st	10
4	Clear City	Anita Desai	1st	10
5	Other Musketeers	Alexander Dumas	1st	10
6	Circle of Reason	Amitav Ghose	1st	10
7	Death of City	Amrita Pritam	1st	10
9	A Secular Agenda	Arun Shourie	1st	10

2.

exam table before updating:

exam_id	subject_id	exam_date	exam_start_at	exam_end_at	class_id
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1	A1	2077-03-14	08:00:00	10:00:00	C01
2	A2	2077-03-15	10:00:00	12:00:00	C02
3	A3	2077-03-17	10:00:00	12:00:00	C07
4	A4	2077-03-18	08:00:00	10:00:00	C02
5	A5	2077-03-19	10:00:00	12:00:00	C10
6	A6	2077-03-22	06:00:00	08:00:00	C09
7	A7	2077-03-23	10:00:00	12:00:00	C06
8	A8	2077-03-24	10:00:00	12:00:00	C07
9	A9	2077-03-25	07:30:00	09:30:00	C02
10	A10	2077-03-26	10:00:00	12:00:00	C03
11	A11	2077-03-27	06:00:00	08:00:00	C03
12	A12	2077-03-28	06:00:00	08:00:00	C17
13	A13	2077-03-29	07:00:00	09:00:00	C18
14	A14	2077-03-30	06:00:00	08:00:00	C17
15	A15	2077-04-01	06:00:00	08:00:00	C17
16	A16	2077-04-02	10:00:00	12:00:00	C18
17	A17	2077-04-03	10:00:00	12:00:00	C19
18	A18	2077-04-05	10:00:00	12:00:00	C03
19	A19	2077-04-06	10:00:00	12:00:00	C02
20	A20	2077-04-07	08:00:00	10:00:00	C15

```sql
UPDATE exam
SET exam_start_at = "07:30:00",
    exam_end_at = "09:30:00"
WHERE exam_id BETWEEN 1 AND 10;
```

exam table after updating:

exam_id	subject_id	exam_date	exam_start_at	exam_end_at	class_id
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1	A1	2077-03-14	07:30:00	09:30:00	C01
2	A2	2077-03-15	07:30:00	09:30:00	C02
3	A3	2077-03-17	07:30:00	09:30:00	C07
4	A4	2077-03-18	07:30:00	09:30:00	C02
5	A5	2077-03-19	07:30:00	09:30:00	C10
6	A6	2077-03-22	07:30:00	09:30:00	C09
7	A7	2077-03-23	07:30:00	09:30:00	C06
8	A8	2077-03-24	07:30:00	09:30:00	C07
9	A9	2077-03-25	07:30:00	09:30:00	C02
10	A10	2077-03-26	07:30:00	09:30:00	C03
11	A11	2077-03-27	06:00:00	08:00:00	C03
12	A12	2077-03-28	06:00:00	08:00:00	C17
13	A13	2077-03-29	07:00:00	09:00:00	C18
14	A14	2077-03-30	06:00:00	08:00:00	C17
15	A15	2077-04-01	06:00:00	08:00:00	C17
16	A16	2077-04-02	10:00:00	12:00:00	C18
17	A17	2077-04-03	10:00:00	12:00:00	C19
18	A18	2077-04-05	10:00:00	12:00:00	C03
19	A19	2077-04-06	10:00:00	12:00:00	C02
20	A20	2077-04-07	08:00:00	10:00:00	C15
21	A21	2077-04-08	06:00:00	08:00:00	C17
22	A22	2077-04-09	07:00:00	09:00:00	C18
23	A23	2077-04-10	06:00:00	08:00:00	C17
24	A24	2077-04-11	06:00:00	08:00:00	C17
25	A25	2077-04-12	10:00:00	12:00:00	C18

3.

staff table before updating:

S20	Aakash Shah	Chef	2057-09-13	NULL	NULL
S21	Minakshat Sharma	Driver	2047-09-13	minak@gmail.com	9840378765
S22	Ram Kharel	Driver	2047-08-13	ramkj12@gmail.com	9840378666
S23	Abishek Upmanyu	Teacher	2056-07-07	abish1@yahoo.com	9840123488
S24	Shaleshma Shah	Cleaning Lady	2047-09-13	singhshal1@gmail.com	9841378095
S25	Roshan Bhardwaj	Warden	2047-05-12	bhardwajr1@gmail.com	9840567834
S26	Deepa Upadhyaye	Teacher	2045-09-08	deepa.uph@yahoo.com	9805630968
S27	Junu Hada	Teacher	2046-06-06	hadajunu1@yahoo.com	9805653890
S28	Arun Basyal	Teacher	2044-02-08	arunbas1@yahoo.com	9808888889
S29	Suyog Amatya	Teacher	2047-08-08	suyog.amatya1@yahoo.co...	9801233211
S30	Deepa Kuikel	Teacher	2053-08-27	deepa.kuikel21@yahoo.co...	9807858034
S31	Pabitra Rai	Teacher	2047-08-08	pabitra.rai12@gmail.com	9845347684

```
UPDATE staff
SET staff_dob = DEFAULT,
    staff_mobile = "9845356380"
WHERE staff_id = 'S31';
```

staff table after updating:

S20	Aakash Shah	Chef	2057-09-13	NULL	NULL
S21	Minakshat Sharma	Driver	2047-09-13	minak@gmail.com	9840378765
S22	Ram Kharel	Driver	2047-08-13	ramkj12@gmail.com	9840378666
S23	Abishek Upmanyu	Teacher	2056-07-07	abish1@yahoo.com	9840123488
S24	Shaleshma Shah	Cleaning Lady	2047-09-13	singhshal1@gmail.com	9841378095
S25	Roshan Bhardwaj	Warden	2047-05-12	bhardwajr1@gmail.com	9840567834
S26	Deepa Upadhyaye	Teacher	2045-09-08	deepa.uph@yahoo.com	9805630968
S27	Junu Hada	Teacher	2046-06-06	hadajunu1@yahoo.com	9805653890
S28	Arun Basyal	Teacher	2044-02-08	arunbas1@yahoo.com	9808888889
S29	Suyog Amatya	Teacher	2047-08-08	suyog.amatya1@yahoo.co...	9801233211
S30	Deepa Kuikel	Teacher	2053-08-27	deepa.kuikel21@yahoo.co...	9807858034
S31	Pabitra Rai	Teacher	NULL	pabitra.rai12@gmail.com	9845356380

4.

entries of staff table having null email before updating:

staff_id	staff_name	staff_job	staff_dob	staff_email	staff_mobile
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
S04	Gajendra Kharel	Cleaner	2041-03-04	NULL	9851345678
S15	Rahul Dulal	Chef	2051-05-14	NULL	9840378712
S19	Hari Kumar	Teacher	2037-12-13	NULL	9840378090
S20	Aakash Shah	Chef	2057-09-13	NULL	NULL

```
UPDATE staff
SET staff_email = "shikshyamail1@gmail.com"
WHERE staff_email is null;
```

entries of staff table having null email after updating:

staff_id	staff_name	staff_job	staff_dob	staff_email	staff_mobile
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
S04	Gajendra Kharel	Cleaner	2041-03-04	shikshyamail1@gmail.com	9851345678
S15	Rahul Dulal	Chef	2051-05-14	shikshyamail1@gmail.com	9840378712
S19	Hari Kumar	Teacher	2037-12-13	shikshyamail1@gmail.com	9840378090
S20	Aakash Shah	Chef	2057-09-13	shikshyamail1@gmail.com	NULL

5.

Some entries of canteen table before updating:

3	Milk tea
2	Sel roti
15	Burger
16	Chana aalu
17	Pizza
7	Cake
2	Lassi

```
UPDATE canteen SET food_name = "Veg Momo"
WHERE student_id =
    (SELECT student_id
    FROM student
    WHERE student_name = "Prakriti Dhakal" );
```

Same entries of canteen table after updating:

3	Milk tea
2	Veg Momo
15	Burger
16	Chana aalu
17	Pizza
7	Cake
2	Veg Momo

6.

Student_transport before updating:

student_id	vehicle_number
Filter...	Filter...
1	Ba 1 Cha 3456
2	Ba 1 Cha 3456
4	Ba 8 Ka 6890
5	Ba 3 Ha 3000
6	Ba 1 Ja 6666
7	Ba 1 Ka 3406
8	Ba 1 Sa 1406
9	Ba 2 Kha 4038
10	Ba 8 Ja 2006
11	Ba 3 Ha 3000
12	Ba 8 Ja 2006
13	Ba 8 Ka 6890
14	Ba 8 Ja 2006
15	Ba 8 Ja 2006

```
UPDATE student_transport SET vehicle_number =
"Ba 1 Cha 3456"
WHERE vehicle_number = "Ba 1 Sa 1406" OR
vehicle_number = "Ba 8 Jha 2006";
```

student_id	vehicle_number
Filter...	Filter...
1	Ba 1 Cha 3456
2	Ba 1 Cha 3456
4	Ba 8 Ka 6890
5	Ba 3 Ha 3000
6	Ba 1 Ja 6666
7	Ba 1 Ka 3406
8	Ba 1 Cha 3456
9	Ba 2 Kha 4038
10	Ba 1 Cha 3456
11	Ba 3 Ha 3000
12	Ba 1 Cha 3456
13	Ba 8 Ka 6890
14	Ba 1 Cha 3456
15	Ba 1 Cha 3456

13. Delete Statement

1.

book_dup before deleting entries:

book_id	book_name	author	edition	quantity
Filter...	Filter...	Filter...	Filter...	Filter...
21	A random book	NULL	1st	10
1	2 states	Chetan Bhagat	1st	10
2	The Three Musketeers	Alexander Dumas	1st	10
3	Development	Amartya Sen	1st	10
4	Clear City	Anita Desai	1st	10
5	Other Musketeers	Alexander Dumas	1st	10
6	Circle of Reason	Amitav Ghose	1st	10
7	Death of City	Amrita Pritam	1st	10
9	A Secular Agenda	Arun Shourie	1st	10

```
DELETE FROM book_dup
WHERE book_id IN (
        SELECT book_id
        FROM book_dup
        WHERE book_name LIKE 'A%'
    );
```

book_dup after deleting entries:

book_id	book_name	author	edition	quantity
Filter...	Filter...	Filter...	Filter...	Filter...
1	2 states	Chetan Bhagat	1st	10
2	The Three Musketeers	Alexander Dumas	1st	10
3	Development	Amartya Sen	1st	10
4	Clear City	Anita Desai	1st	10
5	Other Musketeers	Alexander Dumas	1st	10
6	Circle of Reason	Amitav Ghose	1st	10
7	Death of City	Amrita Pritam	1st	10

2.

top 15 entries of staff_attendance table before deleting:

staff_id	total_days	present_days	absent_days
Filter...	Filter...	Filter...	Filter...
S01	90	90	0
S02	90	85	5
S03	90	80	10
S04	90	85	5
S05	90	84	6
S06	90	79	11
S07	90	86	4
S08	90	80	10
S09	90	84	6
S10	90	80	10
S11	95	88	7
S12	90	85	5
S13	90	70	20
S14	90	70	20
S15	90	80	10

```
DELETE FROM staff_attendance LIMIT 10;
```

top 15 entries of staff_attendance table before deleting:

staff_id	total_days	present_days	absent_days
Filter...	Filter...	Filter...	Filter...
S11	95	88	7
S12	90	85	5
S13	90	70	20
S14	90	70	20
S15	90	80	10
S16	90	85	5
S17	90	88	2
S18	90	89	1
S19	90	79	11
S20	95	80	15
S21	90	82	8
S22	90	85	5
S23	95	88	7
S24	90	75	15
S25	90	90	0

3.

transportation table before deleting

vehicle_number	vehicle_route	vehicle_capacity	transportation_mode	transportation_cost
Filter...	Filter...	Filter...	Filter...	Filter...
Ba 1 Cha 3456	Chabahil - Sankhu	60	Bus	2000.00
Ba 1 Ja 6666	Chabahil - Sanobharyang	30	Van	2000.00
Ba 1 Ka 3406	Chabahil - Maharajgunj	20	Van	2000.00
Ba 1 Sa 1406	Thimi - Bhaktapur	60	Bus	2500.00
Ba 2 Kha 4038	Chabahil - Gothatar	80	Bus	2200.00
Ba 2 Sa 3406	Gaushala - DharmaChowk	60	Bus	2400.00
Ba 3 Ha 3000	Jorpati - Aryalgaun	30	Van	2200.00
Ba 8 Ja 2006	Gaushala - Airport	20	Van	2500.00
Ba 8 Ka 2406	Gaushala - Baneshwor	60	Bus	2000.00
Ba 8 Ka 6890	Chabahil - NMC	60	Bus	2200.00

```
DELETE FROM transportation
WHERE vehicle_number IN (
        SELECT vehicle_number
        FROM transportation
        WHERE transportation_cost = (
                SELECT max(transportation_cost)
                FROM transportation
            )
    );
```

transportation table after deleting:

vehicle_number	vehicle_route	vehicle_capacity	transportation_mode	transportation_cost
Filter...	Filter...	Filter...	Filter...	Filter...
Ba 1 Cha 3456	Chabahil - Sankhu	60	Bus	2000.00
Ba 1 Ja 6666	Chabahil - Sanobharyang	30	Van	2000.00
Ba 1 Ka 3406	Chabahil - Maharajgunj	20	Van	2000.00
Ba 2 Kha 4038	Chabahil - Gothatar	80	Bus	2200.00
Ba 2 Sa 3406	Gaushala - DharmaChowk	60	Bus	2400.00
Ba 3 Ha 3000	Jorpati - Aryalgaun	30	Van	2200.00
Ba 8 Ka 2406	Gaushala - Baneshwor	60	Bus	2000.00
Ba 8 Ka 6890	Chabahil - NMC	60	Bus	2200.00

4.

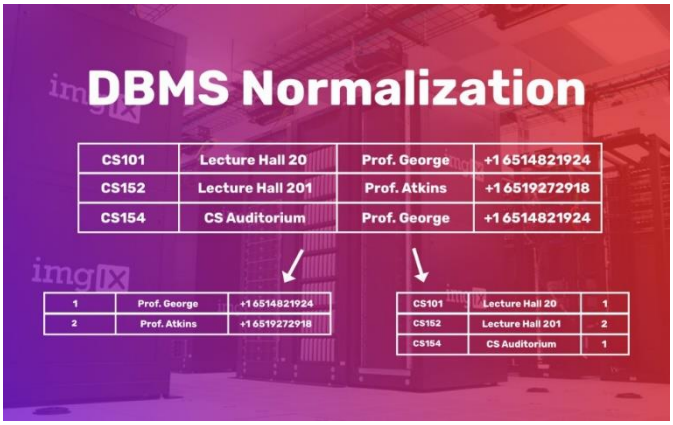
```
DELETE FROM result
WHERE marks_obtained IS NULL;
```

5.

```
DELETE FROM fee
WHERE bill_number NOT IN (
        SELECT bill_number
        FROM fee
        WHERE fee_due <> 0
    );
```

14. Normalization

Database normalization is a technique in database schema design, which modifies an existing scheme to minimize redundancy and dependency of data. Normalization splits a wide table into smaller tables, and identifies relationships between them to improve the clarity of data organization. It was developed by IMB researcher E.F. Codd in the 1970s. Without normalization, many issues like insertion anomaly, update anomaly and deletion anomaly are faced. Normalization in SQL will enhance the distribution of data. (edureka, 2020)



(Goel, 2020)

○ Problems without Normalization

If a table is not properly normalized and has data redundancy, it will consume additional memory space and will be difficult to manage and update the data within the database without the loss of data. Insertion, Updation and Deletion Anomalies are very common if database is not normalized.

- Insertion anomaly: It is the anomaly that occurs when we cannot insert data to the table without the presence of another attribute.
- Update anomaly: It is a data inconsistency that results from data redundancy and a partial update of data.
- Deletion anomaly: It occurs when certain attributes are lost because of the deletion of other attributes.

○ Normalization Rule

Normalization rules are divided into the following normal forms:

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce and Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)

Among these rules, 1NF, 2NF and 3NF are described below:

- First Normal Form (1NF)

  For a table to be in the First Normal Form, it should follow the following 4 rules:

  1. It should only have atomic (single) valued attributes/columns.
  2. Values stored in a column should be of the same domain.
  3. All the columns in a table should have unique names.
  4. And the order in which data is stored, does not matter.

  For example:

Employee ID	Employee Name	Phone Number	Salary
1EDU001	Alex	+91 8553206126 +91 9449424949	60,131
1EDU002	Barry	+91 8762989672	48,302
1EDU003	Clair	+91 9916255225	22,900
1EDU004	David	+91 6363625811 +91 8762055007	81,538

  In this employee table, the Phone Number attribute has multiple values which violates the rules of 1NF. Once we apply the 1NF to the above employee table, we get the below table as the result.

Employee ID	Employee Name	Phone Number	Salary
1EDU001	Alex	+91 8553206126	60,131
1EDU001	Alex	+91 9449424949	60,131
1EDU002	Barry	+91 8762989672	48,302
1EDU003	Clair	+91 9916255225	22,900
1EDU004	David	+91 6363625811	81,538
1EDU004	David	+91 8762055007	81,538

  Thus, atomicity was achieved and every column have unique values.

- Second Normal Form (2NF)

  For a table to be in the Second Normal Form, it should follow the following 2 rules:

  1. It should be in the First Normal Form.
  2. It should not have partial dependency. Here partial dependency means the proper subset of candidate key determines a non-prime attribute.

  For example:

Employee Id	Department Id	Office Location
1EDU001	ED-T1	Pune
1EDU002	ED-S2	Bengaluru
1EDU003	ED-M1	Delhi
1EDU004	ED-T3	Mumbai

  This table has a composite primary key (Employee id and Department id). Here, Office Location is a non-key attribute and it depends only on Department id, which is only a part of the primary key. This clearly violates the second rule of 2NF. Once we apply the 2NF to the above table, we get the below tables as the result.

Employee Id	Department Id
1EDU001	ED-T1
1EDU002	ED-S2
1EDU003	ED-M1
1EDU004	ED-T3

Department Id	Office Location
ED-T1	Pune
ED-S2	Bengaluru
ED-M1	Delhi
ED-T3	Mumbai

  Now, the column Office Location is fully dependent on the primary key of its table, which is Department ID.

- Third Normal Form (3NF)

  For a table to be in the Third Normal Form, it should follow the following 2 rules:

  3. It should be in the Second Normal Form.
  4. It should not have transitive dependency for non-prime attributes. That means non-prime attributes (which doesn't form a candidate key) should not be dependent on other non-prime attributes in a given table.

  For example:

Student Id	Student Name	Subject Id	Subject	Address
1DT15ENG01	Alex	15CS11	SQL	Goa
1DT15ENG02	Barry	15CS13	JAVA	Bengaluru
1DT15ENG03	Clair	15CS12	C++	Delhi
1DT15ENG04	David	15CS13	JAVA	Kochi

  In the above table, Student id determines Subject id, and Subject id determines Subject. Therefore, Student ID determines Subject Via Subject id. This implies that we have a transitive functional dependency and clearly, this does not satisfy the third normal form. Once we apply the 3NF to the above table, we get the below tables as the result.

Student Id	Student Name	Subject Id	Address
1DT15ENG01	Alex	15CS11	Goa
1DT15ENG02	Barry	15CS13	Bengaluru
1DT15ENG03	Clair	15CS12	Delhi
1DT15ENG04	David	15CS13	Kochi

Subject Id	Subject
15CS11	SQL
15CS13	JAVA
15CS12	C++
15CS13	JAVA

  Now, all the non-key attributes are now fully functional dependent only on the primary key. In the first table, columns Student Name, Subject id and Address are only dependent on Student id. In the second table, Subject is only dependent on Subject ID.

15. Conclusion

This report was highly beneficial for understanding the Relational Database Management System (RDBMS) practically. CRUD operations were performed on the database. The research skill was greatly developed through the research that this report required. Basic MySQL queries were used on the database that was created. Furthermore, Information about new MySQL queries that were not on syllabus was researched as well. To sum up, the report was very useful for the basic theoretical and practical understanding of MySQL, its queries and normalization.

16. References

edureka, 2020. *Normalization in SQL.* [Online]
Available at: https://www.edureka.co/blog/normalization-in-sql/
[Accessed 2 June 2020].

Goel, A., 2020. *Normalization in SQL.* [Online]
Available at: https://hackr.io/blog/dbms-normalization/thumbnail/large
[Accessed 2 June 2020].