

TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

MADAN BHANDARI MEMORIAL COLLEGE

New Baneshwor, Kathmandu

**Lab Report of DBMS**

**Submitted by:** **Submitted to:**

Name: Sudip Pradhan Department of B.Sc. CSIT

Symbol No.: 29170

Semester : Fourth

Signature



**Madan Bhandari Memorial College**

**Department of Computer Science and information and technology**

**(B.sc.CSIT)**

**Binayaknagar, New Baneshwor, Kathmandu**

**Practical Record Index**

**Subject : DBMS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name :**Sudip Pradhan | **Semester :** Fourth | **Batch :**  2078 | **Symbol No. :** 29170 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Program No.** | **Title of Programs** | **Date of**  **Submission** | **Signature** |
| **1.** | Installation of MS SQL Server | 2080/11/12 |  |
| **2.** | Create a table and insert values | 2080/11/12 |  |
| **3.** | Create a table and select unique values | 2080/11/12 |  |
| **4.** | Check the working of primary key | 2080/11/12 |  |
| **5.** | Use of primary key and foreign key | 2080/11/18 |  |
| **6.** | Implementation of ALTER | 2080/11/18 |  |
| **7.** | Implementation of DELETE and UPDATE | 2080/11/25 |  |
| **8.** | Altering and adding primary key in a table | 2080/11/25 |  |
| **9.** | Implementation of Group By and Order By | 2080/11/25 |  |
| **10.** | Simple join of two tables | 2080/11/02 |  |
| **11.** | Inner join of two tables | 2080/11/04 |  |
| **12.** | Implementation of primary key and foreign constrains | 2078/11/04 |  |

# Madan Bhandari Memorial College

# 

# Lab No.: 1 Date: 2080/10/12

**Title: Installation fo MS SQL SERVER**

**MS-SQL SERVER**

Microsoft SQL Server (Structured Query Language) is a proprietary relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications

**INSTALLATION STEPS :**

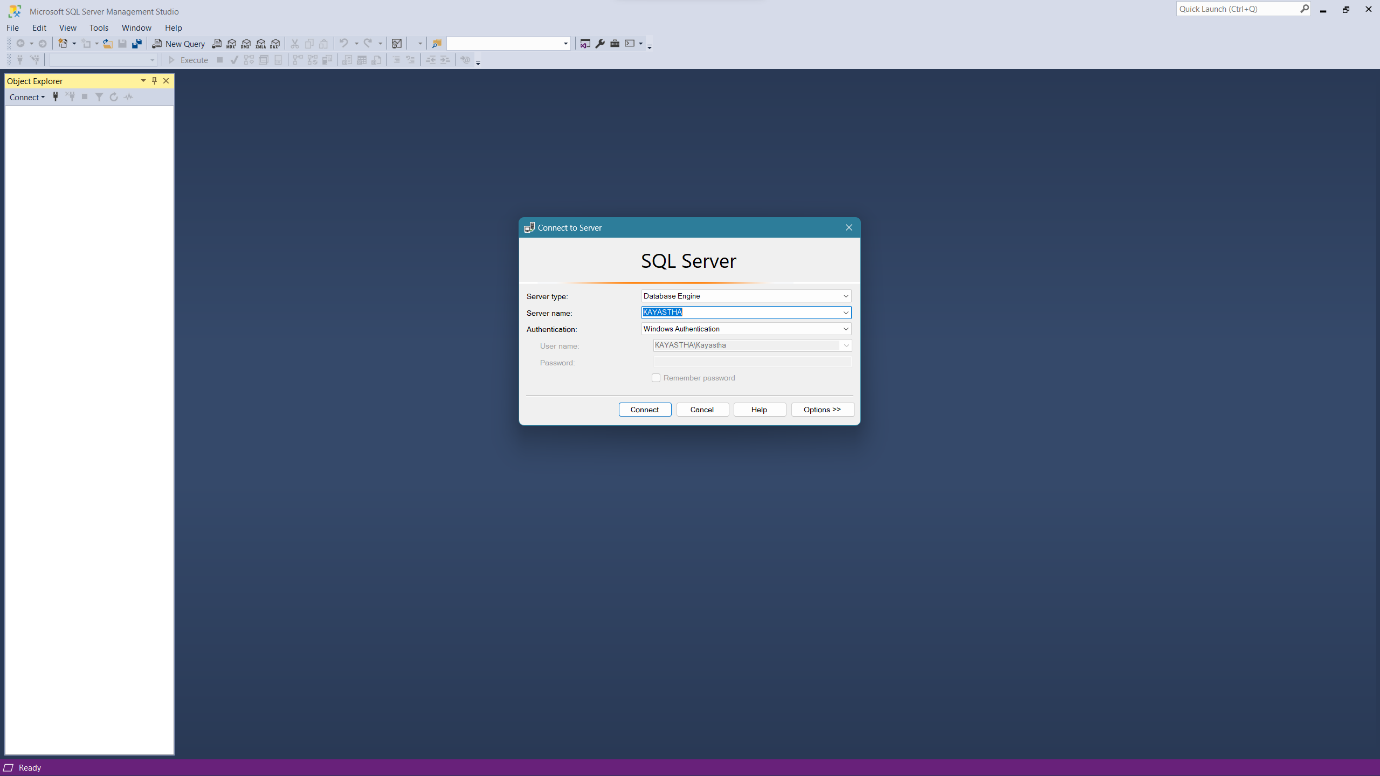
1. Firstly, we download MS-SQL server file.
2. Then, we click on “SQLEXPRA\_X64\_ENU” application to extract “SQLEXPRADV\_X64\_ENU” file.
3. We click on the file and then, click ‘setup’.
4. Click on ‘Installation’ and ‘New SQL server stand-alone’ installation or add features to an existing installation.
5. Then click on, ‘I accept the license term’ and ‘Next’.
6. After clicking few ‘Next’ options, we have to choose ‘Instance Features’. Choose according to the need and ‘Next’.
7. We then choose ‘Default Instance’ and click ‘Next’.
8. After specifying the ‘authentication mode’, click ‘Next’. After clicking, it takes some time to install supporting files and then displays ‘Install Successful’.

Now, We need to install SSMS for connecting SQL Server database.

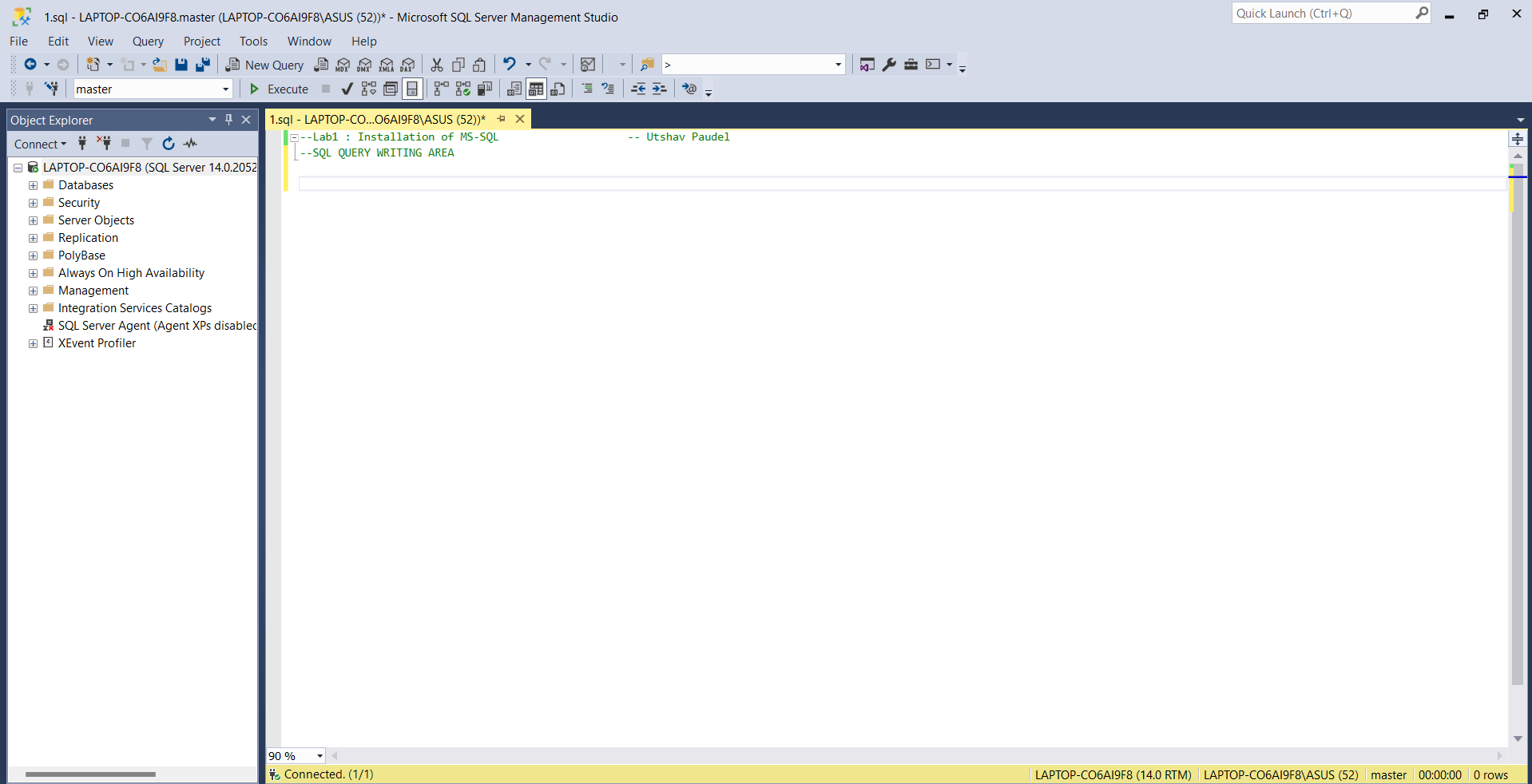
1. Download SQL Server Management Studio (SSMS).
2. Then, click on SSMS-Setup-ENU, choose the location where to install and click ‘INSTALL’.
3. After some time, a window is displayed saying ‘The computer needs to restart before setup can continue’. Click ‘Restart’.
4. After Restart, the file is downloaded. Now click on ‘Microsoft SQL Server Management Studio 18’.
5. After doing installation, we now specify the server type, server name, Authentication and click ‘Connect’.

We have successfully installed both the SQL server and SSMS.

**CONNECTION OF SQL SERVER AND SSMS:**



**CODE WRITING AREA:**





# 

# Lab No.: 2 Date: 2080/10/12

**Title**: Write a SQL code to create a simple table and display its attributes along with its values.

**Theory**:

MS SQL, which stands for Microsoft SQL Server, is a relational database management system (RDBMS) developed by Microsoft. It's a software program that allows you to store, organize, and access data efficiently.

SQL consists of clauses which are instructions within a query that help you specify what data you want to retrieve and how you want it manipulated. Few SQL clauses are:

* SELECT – This clause is the fundamental as it tells SQL what data you want to extract from the database. Asterisk (\*) followed by SELECT will select the entire table.
* FROM – FROM clause specifies the table from which table you want to retrieve.
* WHERE – This clause filters the data based on a specific condition which is specifies by the comparison operators
* CREATE – This clause is used to create table and database
* DROP – This clauses is used to delete the table
* INSERT – This clause is used to insert values in a attributes of table

**Source Code**:

*-- drop table with the similar name if it exists*

drop table if exists practical\_3

*-- create table and give attribute names*

create table practical\_3

(roll\_no int,

name varchar(100),

birth\_date int,

);

-- check if table is creates with the following attributes using select clause

select \* from practical\_3

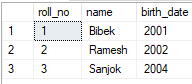
*-- insert data into the table*

insert into practical\_3

values(1, 'Bibek', 2001),

(2, 'Ramesh',2002),

(3, 'Sanjok', 2004);

**Output**

# Madan Bhandari Memorial College

# 

# Lab No.: 3 Date: 2080/10/12

**Title**: Write a SQL code to create a simple table and display the unique values using primary key.

**Theory**:

DISTINCT - This clause is used to display the unique values in an attribute.

**SOURCE CODE:**

*-- drop table if table with similar name exists*

drop table if exists tbl\_Employee

*-- create a table with specific name*

CREATE TABLE tbl\_Employee

(

FirstName varchar(32),

MiddleName varchar(32),

LastName varchar(32),

Age int,

);

*-- insert values into the specified table*

insert into tbl\_Employee(FirstName,MiddleName, LastName, Age)

values('Ram','Prashad', 'Neupane',42),

('Hari','Kumar','Paudel', 33),

('Hari', 'Kumar', 'Paudel', 33);

*-- display everything inside tbl\_Employee*

select \* from tbl\_Employee

*-- display only the unique rows*

select DISTINCT \* from tbl\_Employee

# Output:

# 

# Madan Bhandari Memorial College

# 

# Lab No.: 4 Date: 2080/10/12

**Title**: Write a SQL code and check the use of primary key

**Theory**:

Primary keys and foreign keys are both crucial concepts in relational database design, working together to ensure data integrity and establish relationships between tables. Here's a breakdown of their roles and how they link together.

Primary Key:

* A primary key is a column (or a set of columns) within a table that uniquely identifies each row. It acts as the enforcer of uniqueness, guaranteeing no two rows have the same value for the primary key.
* This ensures efficient data retrieval and manipulation, allowing you to pinpoint specific records without ambiguity.
* Primary keys typically don't allow null values, as a missing value would hinder unique identification.

**SOURCE CODE:**

*-- drop table if table with similar name exists*

drop table if exists tbl\_Employee1

*-- create a table*

create table tbl\_Employee1

(

id int primary key,

FirstName varchar(32) NOT NULL,

MiddleName varchar(32),

LastName varchar(32) not null,

Age int not null,

);

*-- insert values into the table*

insert into tbl\_Employee1(id, FirstName, LastName, Age)

values

(22,'Krishna','Sitaula',44),

(23,'Hari', 'Gupta', 34),

(24, 'Bibek','Thapa', 54),

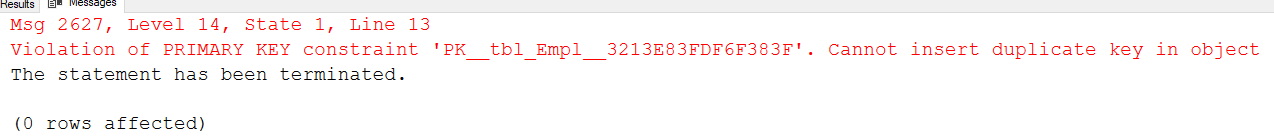
(25, 'Krishna','Bartaula',34);

*-- display the table*

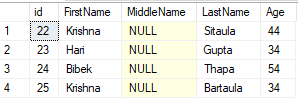
select \* from tbl\_Employee1

**OUTPUT:**

1. When primary key is repeated then it shows “VOILATION OF PRIMARY KEY” which need to be changed.

****

2. When the unique id values are make by correcting the error then the output is:



# Madan Bhandari Memorial College

# 

# Lab No.: 5 Date: 2080/10/18

**Title**: Write a SQL code to create a table and display the use of primary key and foreign key.

**Theory**:

Foreign Key:

* A foreign key resides in a different table than the primary key it references. It creates a link between two tables, establishing a relationship between their data.
* The foreign key column(s) in the child table must contain values that already exist in the primary key column(s) of the referenced parent table.
* This enforces referential integrity, preventing orphaned data (where a child record references a non-existent parent record).

Benefits of Using Primary and Foreign Keys:

* Data Integrity: They prevent inconsistencies and ensure data accuracy by maintaining referential relationships.
* Efficient Data Retrieval: By uniquely identifying rows and linking tables, they streamline queries that involve fetching data from multiple tables.
* Data Modeling: They are fundamental building blocks for creating well-structured and organized relational databases.

**SOURCE CODE:**

*-- 2080/10/18*

*-- lab 6*

*-- drop table if exists*

drop table if exists STD\_ADD

Create table STD\_ADD

(

Roll\_No int primary key,

Names varchar(50),

Address varchar(50),

Place varchar(50),

Pin varchar(50)

);

*-- create table*

CREATE TABLE STD\_MARKS

(

Roll\_No int References STD\_ADD ON DELETE CASCADE,

Subjects varchar(50),

Exam\_date date,

Marks numeric(3)

);

*-- select to see the STD\_ADD table*

select \*

from STD\_ADD

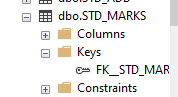
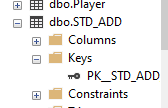
*-- display STD\_MARKS table*

select \*

from STD\_MARKS

**RESULT:**

While using primary key and foreign key, we found that STD\_ADD had a key as a primary key “PK\_STD\_ADD” where as STD\_MARKS table had a key as “FK\_ STD\_MARKS”



# Madan Bhandari Memorial College

# 

# Lab No.: 6 Date: 2080/10/18

**Title**: Write a SQL code to create a table and show the use of ALTER clause.

**Theory:**

The ALTER TABLE clause in SQL is used to modify the structure of an existing table. It's part of the Data Definition Language (DDL) category in SQL and allows you to make various changes to your tables. Here's a breakdown of the ALTER TABLE clause and its functionalities:

Basic Syntax:

ALTER TABLE [Table Name]

[ALTER COLUMN | ADD COLUMN | DROP COLUMN | RENAME TO | ...];

Explanation of Components:

* ALTER TABLE [Table Name]: This specifies the name of the table you want to modify.
* The clause following ALTER TABLE: This defines the specific modification you want to make to the table. Here are some common options:
* ALTER COLUMN: Modify an existing column's data type, constraints, or nullability.
* ADD COLUMN: Add a new column to the table, specifying its name, data type, and any constraints.
* DROP COLUMN: Remove an existing column from the table.
* RENAME TO: Rename the table itself or a specific column within the table.

**SOURCE CODE:**

*-- drop table if table with similar name exists*

drop table if exists Lab7

*-- create a table as Lab7*

create table Lab7

(roll\_no int,

name varchar(50),

birth\_date int

);

insert into Lab7 Values

(03, 'Aayushmi', 2004),

(13, 'Salini', 2002),

(19, 'Sudip', 2001);

*-- check the table with display clause i.e select*

select \* from Lab7

*-- use alter clause which add the new column as Email*

ALTER Table Lab7

add Email varchar(50)

*-- insert the values inside the table*

insert into Lab7 Values

(03, 'Aayushmi', 2004,'aayushmi@gmail.com'),

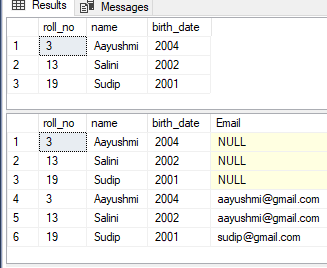
(13, 'Salini', 2002, 'aayushmi@gmail.com'),

(19, 'Sudip', 2001, 'sudip@gmail.com');

*-- display the table along with email attribute column*

select \* from Lab7

**OUTPUT:**

****

# Madan Bhandari Memorial College

# 

# Lab No.: 7 Date: 2080/10/25

**Title**: Write a SQL code to create a table and show the use of DELETE and UPDATE clauses.

**Theory:**

DELETE Clause: The DELETE clause permanently removes rows from a table.

* Syntax:

DELETE FROM [Table Name]

[WHERE Clause];

UPDATE Clause: The UPDATE clause modifies existing data within a table.

* Syntax:

UPDATE [Table Name]

SET [Column Name 1] = [New Value 1],

[Column Name 2] = [New Value 2], ...

[WHERE Clause];

**SOURCE CODE:**

*-- Lab 9: 2080/10/25*

*-- drop database if exists Employee*

*--create database Employee*

use Employee

drop table if exists employee\_tab

create table employee\_tab

(

EmpId int primary key,

FirstName varchar(50),

LastName varchar(50),

Salary int,

Mobile varchar(50)

);

insert into employee\_tab values

(1001, 'Hari', 'Lamsal', 100000, 9834568234),

(1002, 'Binod','Jha', 10000, 9865783456),

(1003, 'Ram','Gupta', 433000, 9823463578)

select \* from employee\_tab

delete from employee\_tab where EmpId = 1003

select \* from employee\_tab

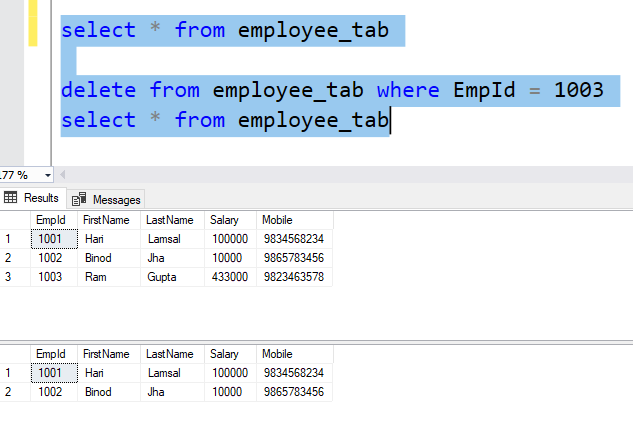
update employee\_tab

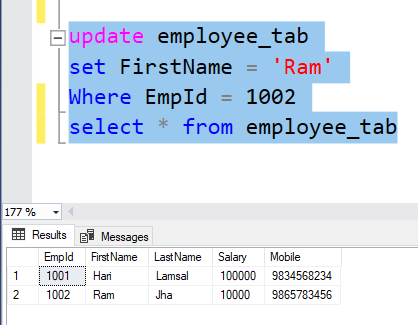
set FirstName = 'Ram'

Where EmpId = 1002

select \* from employee\_tab

**OUTPUT:**

****

****

# Madan Bhandari Memorial College

# 

# Lab No.: 8 Date: 2080/10/25

**Title**: ALTERING AND ADDING PRIMARY KEY IN TABLE

**SOURCE CODE:**

drop table if exists Supplier1

create table Supplier1

(

ID int,

Name varchar(50),

S\_code int,

Deposit int

)

select \* from Supplier1

alter table Supplier1

add PINCODE int, city varchar(50)

select \* from Supplier1

alter table Supplier1

Add ID1 int primary key

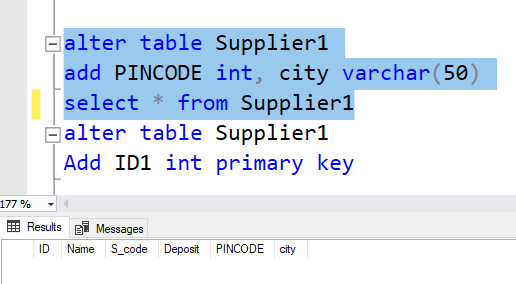
select \*from Supplier1

**OUTPUT:**

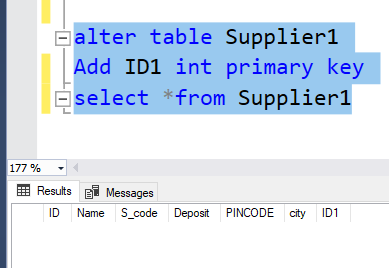
Before altering or adding city and PIN CODE

****

After adding PINCODE and City

****

After adding ID1 as primary key

****

# Madan Bhandari Memorial College

# Lab No.: 9 Date: 2080/10/25

**TITLE: IMPLEMENTATION OF GROUP BY AND ORDER BY**

**SOURCE CODE:**

drop table if exists customerIO

create table CUSTOMERIO

(CUST\_NO numeric(4) primary key,

LNAME varchar(10),

FNAME varchar(15),

ADDR varchar(20),

CITY varchar(20),

STATES varchar(10),

PIN varchar(3),

BIRTH\_DATE date,

);

select \* from CUSTOMERIO;

insert into CUSTOMERIO

values(0001,'pradhan','ram','baneshwor','ktm','bagmati','342','2060/10/10'),

(0002,'pradhan','sam','baneshwor','ktm','bagmati','342','2050/10/10'),

(34,'pradhan','ram','baneshwor','ktm','gandaki','342','2060/12/10'),

(0041,'pradhan','hari','baneshwor','ktm','karnali','342','2065/10/10'),

(00051,'pradhan','sandeep','baneshwor','ktm','lumbini','342','2060/10/10')

*--conditional statement*

select \* from CUSTOMERIO

where STATES = 'bagmati';

*--order by*

select \* from CUSTOMERIO

order by FNAME;

*--DESCENDING ORDER*

select \* from CUSTOMERIO

order by FNAME desc;

*-- ASCENDING ORDER*

select \* from CUSTOMERIO

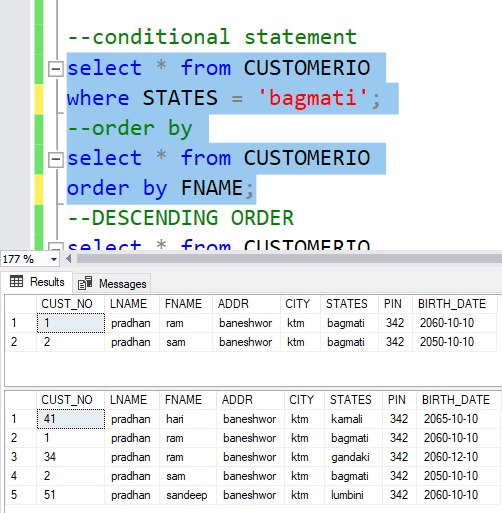
order by FNAME asc;

*--sort by lname*

select \* from CUSTOMERIO

order by FNAME, LNAME ;

**OUTPUT:**

****

# Madan Bhandari Memorial College

# Lab No.: 10 Date: 2080/10/25

**Title: Simple Join**

**SOURCE CODE:**

drop table if exists player

create table player

(

ROLLNO numeric(5),

NAMES varchar(10)

);

--inserting values

insert into player(ROLLNO, NAMES)

values(10,'Ram'),

(20,'Hari'),

(30,'Rita'),

(40,'Gita');

select \* from player;

drop table if exists matches

create table matches

(

MATCHNO numeric(5),

ROLLNO numeric(5),

MATCH\_DATE DATE,

OPPONENT varchar(10)

);

insert into matches(MATCHNO,ROLLNO,MATCH\_DATE,OPPONENT)

values(1,20,'10-JAN-2009','RAJIV'),

(2,30,'10-FEB-2009','RAHUL'),

(3,20,'12-FEB-2009','RAJAT'),

(4,39,'10-JAN-2009', 'MAYUR');

select\* from matches

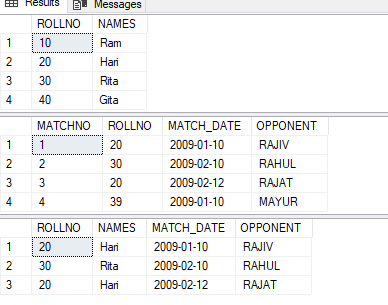
--INNER JOIN

SELECT player.ROLLNO, NAMES, MATCH\_DATE, OPPONENT

FROM player , matches

where player.ROLLNO = matches.ROLLNO;

**OUTPUT:**

****

# Madan Bhandari Memorial College

# Lab No.: 11 Date: 2080/10/25

**Title: Inner Join**

**SOURCE CODE:**

drop table if exists tab1

create table tab1

(

Num1\_ID int,

);

*-- inserting value into first table*

insert into tab1(Num1\_ID)

values(12), (14), (10), (11)

*--creating second table*

drop table if exists tab2

create table tab2

(

Num2\_ID int,

);

*--inserting values into second table*

insert into tab2(Num2\_ID)

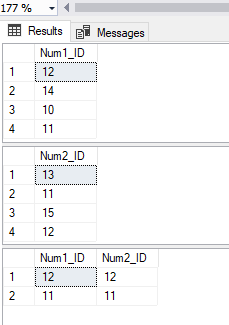
values (13), (11), (15), (12)

select \* from tab1

select \* from tab2

select \* from tab1 Inner Join tab2 ON tab1.Num1\_ID = tab2.Num2\_ID

**OUTPUT:**

****



# Lab No.: 12 Date: 2080/10/25

**TITLE: IMPLEMENTATION OF PRIMARY KEY AND FOREIGN KEY CONSTRAINS**

**SOURCE CODE:**

create table customers(

ID int not null,

Name varchar(40),

Email varchar(40),

Payment varchar(40),

primary key(ID), *--- primary key*

);

create table orders(

ID int not null,

OrderDescription varchar(40),

OrderDate varchar(40),

Price int,

primary key(ID),

Customer\_ID int references customers(ID), *--- foreign key*

);

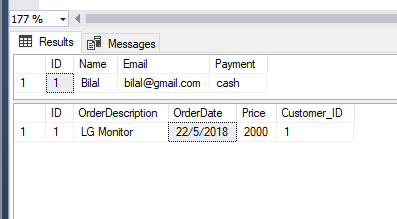
insert into customers values(1,'Bilal','bilal@gmail.com','cash');

insert into orders values(1,'LG Monitor','22/5/2018',2000,1);

select \* from customers

select \* from orders

**OUTPUT:**

****