Lab 10:

Create table sales and products using key constraints (primary and foreign key), insert suitable data and perform DML operation (select clause, where clause and aggregate functions).

SQL Queries:

- CREATE DATABASE anshu;
- USE anshu;
- CREATE TABLE products (
  PID INT PRIMARY KEY,
  P_Name VARCHAR(100) ,
  Price DECIMAL(10, 2)
  );

- CREATE TABLE Sales (
  SID INT PRIMARY KEY,
  PID INT,
  Quantity INT,
  SaleDate DATE,
  FOREIGN KEY (PID) REFERENCES products(PID)
  );

- INSERT INTO products (PID, P_Name, Price) VALUES
  (1, 'Laptop', 120000),
  (2, 'Smartphone', 8000.00),
  (3, 'Tablet', 40000.00),
  (4, 'Monitor', 25000),
  (5, 'Keyboard', 500);

PID	PName	Price
1	Laptop	120000.00
2	Smartphone	8000.00
3	Tablet	40000.00
4	Monitor	25000.00
5	Keyboard	500.00

- INSERT INTO sales (SID, PID, Quantity, SaleDate) VALUES
  (101, 1, 2, '2025-08-01'),
  (102, 2, 3, '2025-08-02'),
  (103, 1, 1, '2025-08-03'),
  (104, 3, 4, '2025-08-03'),
  (105, 5, 10, '2025-08-04'),
  (106, 4, 1, '2025-08-05');

SID	PID	Quantity	SaleDate
101	1	2	2025-08-01
102	2	3	2025-08-02
103	1	1	2025-08-03
104	3	4	2025-08-03
105	5	10	2025-08-04
106	4	1	2025-08-05

- To view sales with product details

```
SELECT
S.SID,
P.P_Name,
S.Quantity,
P.Price,
 (S.Quantity * P.Price) AS TotalSaleAmount,
S.SaleDate
FROM sales S
JOIN products P ON S.PID = P.PID;
```

SID	PName	Quantity	Price	TotalSaleAmount	SaleDate
101	Laptop	2	120000.00	240000.00	2025-08-01
102	Smartphone	3	8000.00	24000.00	2025-08-02
103	Laptop	1	120000.00	120000.00	2025-08-03
104	Tablet	4	40000.00	160000.00	2025-08-03
105	Keyboard	10	500.00	5000.00	2025-08-04
106	Monitor	1	25000.00	25000.00	2025-08-05

- To filter sales for a specific product using WHERE clause

```
SELECT
S.SID,
 P.P_Name,
 S.Quantity,
S.SaleDate
FROM Sales S
JOIN products P ON S.PID = P.PID
WHERE P.P_Name = 'Laptop';
```

SID	PName	Quantity	SaleDate
101	Laptop	2	2025-08-01
103	Laptop	1	2025-08-03

- To calculate total revenue and average quantity sold

```
SELECT
SUM(S.Quantity * P.Price) AS TotalRevenue
FROM sales S
JOIN products P ON S.PID = P.ID;
```

TotalRevenue
574000.00

```
SELECT
AVG(Quantity) AS AvgQuantityPerSale
FROM Sales;
```

AvgQuantityPerSale
3.5000

- To count total sales for each product

```
SELECT
P.P_Name,
 COUNT(S.SaleID) AS TotalSales
FROM sales S
JOIN products P ON S.PID = P.PID
GROUP BY P.P_Name;
```

PName	TotalSales
Keyboard	1
Laptop	2
Monitor	1
Smartphone	1
Tablet	1

Lab 9:

Create table student with suitable attributes and insert 10-15 records and perform the matching operation using LIKE keywords and use all given char functions (substring, concat, length, upper, lower, trim, ltrim, rtrim, char, ascii).

SQL Queries:

- CREATE DATABASE Anshu;
- USE Anshu;
- CREATE TABLE student(

  eid INT(3),

  ename VARCHAR(20),

  eaddress VARCHAR(20),

  ephone NUMERIC(10),

  eage INT(3)

  );

- INSERT INTO student VALUES

  (1, ' Anshu Hada ', ' Dhobidhara ', 9812345601, 22),

  (2, ' Barsha Pandey ', ' Dillibazar ', 9812345602, 25),

  (3, ' Gaurav Thapa ', ' Baneshwor', 9812345603, 30),

  (4, ' Manila Aryal', 'Kalanki', 9812345604, 28),

  (5, 'Kamana Shrestha ', ' Putalisadak ', 9812345605, 22),

  (6, ' Shrisha Tuladhar ', ' Kirtipur', 9812345606, 24),

  (7, ' Krisma Maharjan', 'Gausala ', 9812345607, 31),

  (8, ' Sudip Khadka ', ' Maitidevi', 9812345609, 29),

  (9, 'Ayush Tuladhar ', 'Dallu ', 9812345609, 26),

  (10, ' Sangam Adhikari ', ' Sankhamul ', 9812345610, 23),

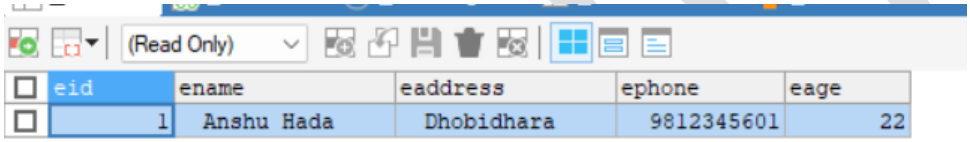
  (11, ' Abhilekh Sudebi', ' Gongabu', 9812345611, 32),

  (12, 'Hrikesh Aran ', ' Balaju ', 9812345612, 30);

  SELECT * FROM student;
- SELECT * FROM student WHERE ename LIKE '%a%';

eid	ename	eaddress	ephone	eage
1	Anshu Hada	Dhobidhara-30	9812345601	22
2	Barsha Pandey	Dillibazar-5	9812345602	25
3	Gaurav Thapa	Baneshwor-2	9812345603	30
4	Manila Aryal	Kalanki-1	9812345604	28
5	Kamana Shrestha	Putalisadak-4	9812345605	22
6	Shrisha Tuladhar	Kirtipur-6	9812345606	24
7	Krisma Maharjan	Gausala-7	9812345607	31
8	Sudip Khadka	Maitidevi-2	9812345609	29
9	Ayush Tuladhar	Balkot-3	9812345609	26
10	Sangam Adhikari	Sankhamul-9	9812345610	23
11	Abhilekh Sudebi	Gongabu-4	9812345611	32
12	Hrikesh Aran	Balaju-10	9812345612	30
13	Upendra Panta	Samakhushi-5	9812345613	33
14	Anush Shrestha	New Baneshwor-8	9812345614	28
15	Ashlesha Shrestha	Jorpati-3	9812345615	27

- SELECT *FROM student WHERE eaddress LIKE '%Dhobidhara%';



eid	ename	eaddress	ephone	eage
1	Anshu Hada	Dhobidhara	9812345601	22

- SELECT *FROM student WHERE ephone LIKE '%2';

eid	ename	eaddress	ephone	eage
2	Barsha Pandey	Dillibazar	9812345602	25
12	Hrikesh Aran	Balaju	9812345612	30

- SELECT eid, SUBSTRING(ename,1,5) AS Name5 FROM student WHERE eid<5;

eid	Name5
1	Ans
2	Bars
3	Gaur
4	Man

- SELECT CONCAT(eid, ' ', ename) AS primarykey FROM student;

primarykey
1 Anshu Hada
2 Barsha Pandey
3 Gaurav Thapa
4 Manila Aryal
5 Kamana Shrestha
6 Shrisha Tuladhar
7 Krisma Maharjan
8 Sudip Khadka
9 Ayush Tuladhar
10 Sangam Adhikari
11 Abhilekh Sudebi
12 Hrikesh Aran

- SELECT eaddress, LENGTH(eaddress) AS Addresslength FROM student;

eaddress	Addresslength
Dhobidhara	14
Dillibazar	13
Baneshwor	11
Kalanki	8
Putalisadak	15
Kirtipur	9
Gausala	8
Maitidevi	11
Dallu	7
Sankhamul	10
Gongabu	8
Balaju	9

- SELECT ename, UPPER(ename) AS UpperName, LOWER(ename) AS LowerName FROM student;

ename	UpperName	LowerName
Anshu Hada	ANSHU HADA	anshu hada
Barsha Pandey	BARSHA PANDEY	barsha pandey
Gaurav Thapa	GAURAV THAPA	gaurav thapa
Manila Aryal	MANILA ARYAL	manila aryal
Kamana Shrestha	KAMANA SHRESTHA	kamana shrestha
Shrisha Tuladhar	SHRISHA TULADHAR	shrisha tuladhar
Krisma Maharjan	KRISMA MAHARJAN	krisma maharjan
Sudip Khadka	SUDIP KHADKA	sudip khadka
Ayush Tuladhar	AYUSH TULADHAR	ayush tuladhar
Sangam Adhikari	SANGAM ADHIKARI	sangam adhikari
Abhilekh Sudebi	ABHILEKH SUDEBI	abhilekh sudebi
Hrikesh Aran	HRIKESH ARAN	hrikesh aran

- SELECT

  TRIM(ename) AS TrimmedName,

  LTRIM(eaddress) AS LeftTrimmed,

  RTRIM(ename) AS RightTrimmed

  FROM student;

TrimmedName	LeftTrimmed	RightTrimmed
Anshu Hada	Dhobidhara	Anshu Hada
Barsha Pandey	Dillibazar	Barsha Pandey
Gaurav Thapa	Baneshwor	Gaurav Thapa
Manila Aryal	Kalanki	Manila Aryal
Kamana Shrestha	Putalisadak	Kamana Shrestha
Shrisha Tuladhar	Kirtipur	Shrisha Tuladhar
Krisma Maharjan	Gausala	Krisma Maharjan
Sudip Khadka	Maitidevi	Sudip Khadka
Ayush Tuladhar	Dallu	Ayush Tuladhar
Sangam Adhikari	Sankhamul	Sangam Adhikari
Abhilekh Sudebi	Gongabu	Abhilekh Sudebi
Hrikesh Aran	Balaju	Hrikesh Aran

- SELECT ename, ASCII(SUBSTRING(LTRIM(ename), 1, 1)) AS firstcharASCII FROM student;

ename	firstcharASCII
Anshu Hada	65
Barsha Pandey	66
Gaurav Thapa	71
Manila Aryal	77
Kamana Shrestha	75
Shrisha Tuladhar	83
Krisma Maharjan	75
Sudip Khadka	83
Ayush Tuladhar	65
Sangam Adhikari	83
Abhilekh Sudebi	65
Hrikesh Aran	72

- SELECT ename, eage, CHAR(ASCII(SUBSTRING(CAST(eage AS CHAR), 1, 1))) AS FirstChar FROM student;

ename	eage	FirstChar
Anshu Hada	22	2
Barsha Pandey	25	2
Gaurav Thapa	30	3
Manila Aryal	28	2
Kamana Shrestha	22	2
Shrisha Tuladhar	24	2
Krisma Maharjan	31	3
Sudip Khadka	29	2
Ayush Tuladhar	26	2
Sangam Adhikari	23	2
Abhilekh Sudebi	32	3
Hrikesh Aran	30	3

Lab 11:

a) Write sql code to create alias name of existing attributes.
b) Create table teacher with suitable fields
c) Insert seven records.
d) Give increment of 30% salary of computer department.
e) Give increment of 50% of salary who works more than 10 years.
f) Find the highest paying and lowest paying teacher from math department

SQL Queries:

- CREATE database ANU;
- USE ANU;
- CREATE TABLE teacher (
  teacher_id INT PRIMARY KEY,
  teacher_name VARCHAR(100),
  department VARCHAR(50),
  salary DECIMAL(10, 2),
  years_of_experience INT
  );
- INSERT INTO teacher (teacher_id, teacher_name, department, salary, years_of_experience) VALUES
  (1, 'Anshu Hada', 'English', 50000, 12),
  (2, 'Prashant Karki', 'Computer', 60000, 9),
  (3, 'Aradhya Neupane', 'Math', 55000, 7),
  (4, 'Riya Shahi', 'Science', 62000, 11),
  (5, 'Diya Tamang', 'Physics', 52000, 15),
  (6, 'Aryan Shrestha', 'Nepali', 58000, 4),
  (7, 'Prabin Khadka', 'Math', 49000, 10);
  select * from teacher ;

teacher_id	teacher_name	department	salary	years_of_experience
1	Anshu Hada	English	50000.00	12
2	Prashant Karki	Computer	60000.00	9
3	Aradhya Neupane	Math	55000.00	7
4	Riya Shahi	Science	62000.00	11
5	Diya Tamang	Physics	52000.00	15
6	Aryan Shrestha	Nepali	58000.00	4
7	Prabin Khadka	Math	49000.00	10

- To give increment of 30% of salary of computer department.

  UPDATE teacher

  SET salary = salary * 1.30

  WHERE department = 'Math';

teacher_id	teacher_name	department	salary	years_of_experience
1	Anshu Hada	English	50000.00	12
2	Prashant Karki	Computer	60000.00	9
3	Aradhya Neupane	Math	71500.00	7
4	Riya Shahi	Science	62000.00	11
5	Diya Tamang	Physics	52000.00	15
6	Aryan Shrestha	Nepali	58000.00	4
7	Prabin Khadka	Math	63700.00	10

- To give increment of 50% of salary who works more than 10 years.

  UPDATE teacher

  SET salary = salary * 1.50

  WHERE years_of_experience > 10;

teacher_id	teacher_name	department	salary	years_of_experience
1	Anshu Hada	English	75000.00	12
2	Prashant Karki	Computer	60000.00	9
3	Aradhya Neupane	Math	71500.00	7
4	Riya Shahi	Science	93000.00	11
5	Diya Tamang	Physics	78000.00	15
6	Aryan Shrestha	Nepali	58000.00	4
7	Prabin Khadka	Math	63700.00	10

- To find the highest paying and lowest paying teacher from math department

  SELECT *

  FROM teacher

  WHERE department = 'Math'

AND salary = (SELECT MAX(salary) FROM teacher WHERE department = 'Math');

teacher_id	teacher_name	department	salary	years_of_experience
3	Aradhya Neupane	Math	71500.00	7

SELECT *

FROM teacher

WHERE department = 'Math'

AND salary = (SELECT MIN(salary) FROM teacher WHERE department = 'Math');

teacher_id	teacher_name	department	salary	years_of_experience
3	Aradhya Neupane	Math	71500.00	7

- To create alias name of existing attributes.

SELECT

teacher_id AS ID,

teacher_name AS NAME,

department AS Dept,

salary AS Salary,

years_of_experience AS Experience

FROM teacher;

ID	NAME	Dept	Salary	Experience
1	Anshu Hada	English	75000.00	12
2	Prashant Karki	Computer	60000.00	9
3	Aradhya Neupane	Math	71500.00	7
4	Riya Shahi	Science	93000.00	11
5	Diya Tamang	Physics	78000.00	15
6	Aryan Shrestha	Nepali	58000.00	4
7	Prabin Khadka	Math	63700.00	10

LAB 8: Write SQL code to display records using cross, equi and self join.

SQL Queries:

- CREATE DATABASE lab8;
- USE lab8;
- CREATE TABLE department (
  department_id INT PRIMARY KEY,
  department_name VARCHAR(50)
  );
- CREATE TABLE employee (
  emp_id INT PRIMARY KEY,
  NAME VARCHAR(50),
  department_id INT,
  manager_id INT

);

- INSERT INTO department (department_id, department_name) VALUES
  (10, 'HR'),
  (20, 'IT'),
  (30, 'Sales');

department_id	department_name
10	HR
20	IT
30	Sales

- INSERT INTO employee (emp_id, NAME, department_id, manager_id) VALUES
  (1, 'Anshu Hada', 10, NULL),
  (2, 'Gaurav Thapa', 20, 1),
  (3, 'Sudip Khadka', 10, 1),
  (4, 'Anush Shrestha', 30, 2);

emp_id	NAME	department_id	manager_id
1	Anshu Hada	10	*(NULL)*
2	Gaurav Thapa	20	1
3	Sudip Khadka	10	1
4	Anush Shrestha	30	2

- Displaying data using Cross join:

  SELECT e.name AS employee_name, d.department_name

  FROM employee e

  CROSS JOIN department d;

employee_name	department_name
Anshu Hada	HR
Anshu Hada	IT
Anshu Hada	Sales
Gaurav Thapa	HR
Gaurav Thapa	IT
Gaurav Thapa	Sales
Sudip Khadka	HR
Sudip Khadka	IT
Sudip Khadka	Sales
Anush Shrestha	HR
Anush Shrestha	IT
Anush Shrestha	Sales

- Displaying data using Equi join.

  SELECT e.name AS employee_name, d.department_name FROM employee e

  JOIN department d ON e.department_id = d.department_id;

employee_name	department_name
Anshu Hada	HR
Gaurav Thapa	IT
Sudip Khadka	HR
Anush Shrestha	Sales

- Displaying data using Self join.

  SELECT e.name AS employee_name, m.name AS manager_name FROM employee e

  JOIN employee m ON e.manager_id = m.emp_id;

employee_name	manager_name
Gaurav Thapa	Anshu Hada
Sudip Khadka	Anshu Hada
Anush Shrestha	Gaurav Thapa