Lab Manuals Name: Implementation of Relational Databases (Join Function)

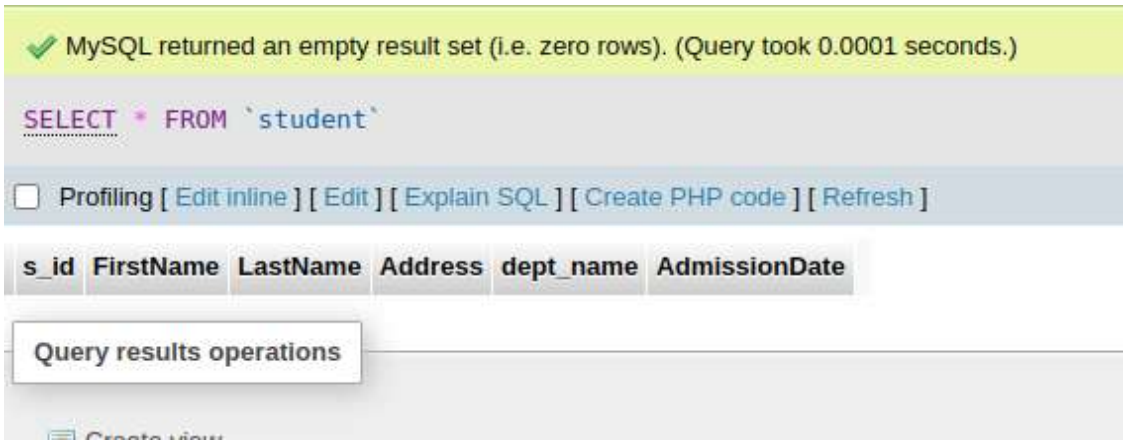
30.11.2024

—

Promod Chandra Das
Dept: CSE
Id: 231002005

• Create a table student:

```
CREATE TABLE student (
    s_id INT(11) NOT NULL AUTO_INCREMENT,
    FirstName VARCHAR(255) NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    dept_name ENUM('CSE', 'EEE', 'TEXT') DEFAULT NULL,
    AdmissionDate DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP(),
    PRIMARY KEY(s_id)
);
```



2. INSERT INTO student (s_id, FirstName, LastName, Address, dept_name, AdmissionDate) VALUES

(1, 'Zeseya', 'Sharmin', 'Dhaka', 'CSE', '2024-11-25 12:00:00'),

(2, 'Sakib', 'Hasan', 'Natore', 'CSE', '2024-11-25 15:00:00'),

(3, 'Asef', 'Tajwar', 'Rangpur', 'EEE', '2024-11-25 16:00:00'),

(4, 'Maruf', 'Hasan', 'Barisal', 'EEE', '2024-11-25 17:00:00'),

(5, 'Ashek', 'Farabi', 'Gazipur', 'TEX', '2024-11-25 18:00:00'),

(6, 'Ismile', 'Hasan', 'Barisal', 'TEX', '2024-11-25 19:00:00');

Extra options

	s_id	FirstName	LastName	Address	dept_name	AdmissionDate
Edit Copy Delete	1	Zeseya	Sharmin	Dhaka	CSE	2024-11-25 12:00:00
Edit Copy Delete	2	Sakib	Hasan	Natore	CSE	2024-11-25 15:00:00
Edit Copy Delete	3	Asef	Tajwar	Rangpur	EEE	2024-11-25 16:00:00
Edit Copy Delete	4	Maruf	Hasan	Barisal	EEE	2024-11-25 17:00:00
Edit Copy Delete	5	Ashek	Farabi	Gazipur	TEX	2024-11-25 18:00:00
Edit Copy Delete	6	Ismile	Hasan	Barisal	TEX	2024-11-25 19:00:00

3. CREATE TABLE department (

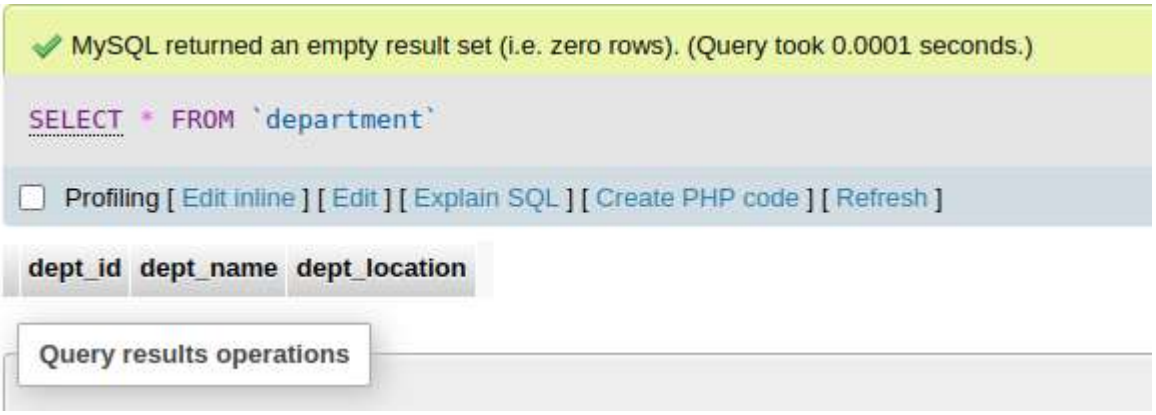
dept_id INT(11) NOT NULL AUTO_INCREMENT,

dept_name ENUM('CSE', 'EEE', 'TEX') DEFAULT NULL,

dept_location VARCHAR(255) NOT NULL,

PRIMARY KEY(dept_id)

);



4. INSERT INTO department (dept_id, dept_name, dept_location)
VALUES
(101, 'CSE', 'Building-2'),
(102, 'EEE', 'Building-2'),
(103, 'TEX', 'Building-1');

Extra options

	dept_id	dept_name	dept_location
Edit Copy Delete	101	CSE	Building-2
Edit Copy Delete	102	EEE	Building-2
Edit Copy Delete	103	TEX	Building-1

5. CREATE TABLE course_registration (

reg_serial INT(11) NOT NULL AUTO_INCREMENT,

course_code VARCHAR(255) NOT NULL,

course_title VARCHAR(255) NOT NULL,

dept_id INT(11) NOT NULL,

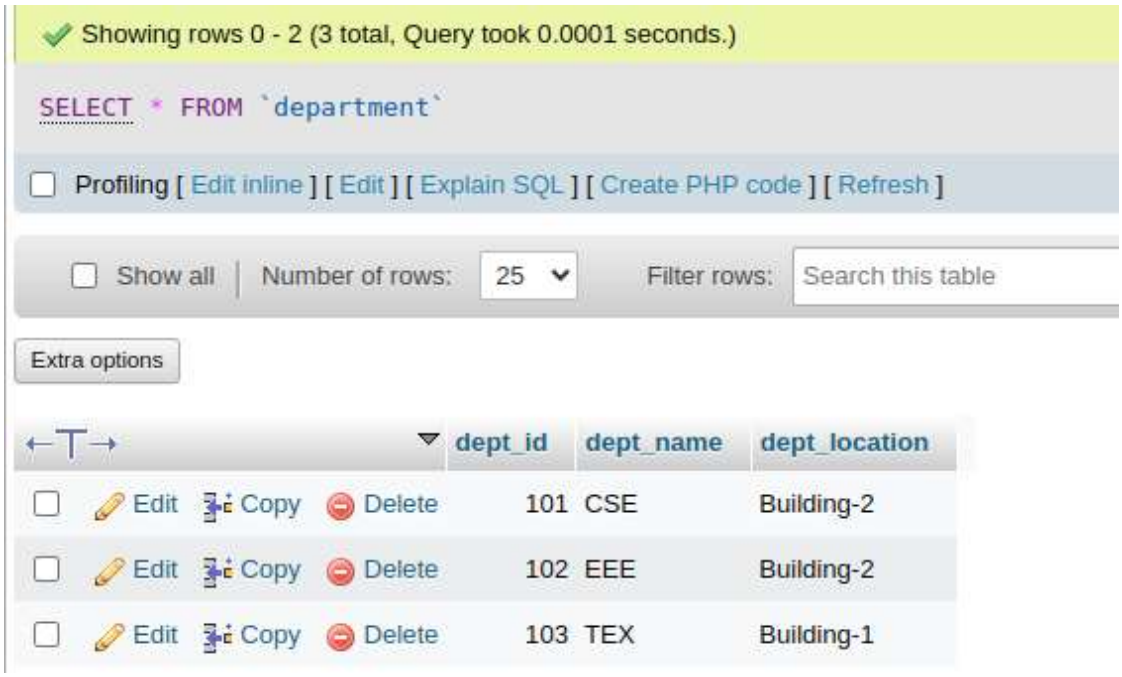
s_id INT(11) NOT NULL, -- Assuming student ID is an integer

PRIMARY KEY (reg_serial),

FOREIGN KEY (dept_id) REFERENCES department(dept_id),

FOREIGN KEY (s_id) REFERENCES student(s_id)

);


Showing rows 0 - 2 (3 total, Query took 0.0001 seconds.)
SELECT * FROM `department`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Extra options
dept_id
dept_name
dept_location
Edit Copy Delete 101 CSE Building-2
Edit Copy Delete 102 EEE Building-2
Edit Copy Delete 103 TEX Building-1

6.SELECT * FROM student WHERE s_id IN (142002015, 142002001, 162002002, 172002002);

INSERT INTO course_registration(course_code,Course_title,dept_id,s_id)

VALUES ('CSE 311', 'Computer Networks',101,142002015),

('CSE 311', 'Computer Networks',101,142002001),

('EEE 301','Electrical Circuit',201,162002002),

('TEX 201', 'Aparales', 301,172002002),

('CSE 312', 'Computer Networks Lab',101,142002015),

('CSE 207', 'Algorithm',101,142002001);

7. SELECT s_id
FROM student
UNION
SELECT s_id
FROM course_registration;

8. SELECT s_id

FROM student

UNION ALL

SELECT s_id

FROM course_registration;

Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

```
SELECT s_id FROM student UNION ALL SELECT s_id FROM course_registration;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

_id
1
2
3
4
5

8. SELECT student.s_id, student.FirstName, student.LastName

FROM student

INNER JOIN course_registration ON student.s_id = course_registration.s_id;

	s_id	FirstName	LastName	Address	dept_name	AdmissionDate
Edit Copy Delete	1	Zeseya	Sharmin	Dhaka	CSE	2024-11-25 12:00:00
Edit Copy Delete	2	Sakib	Hasan	Natore	CSE	2024-11-25 15:00:00
Edit Copy Delete	3	Asef	Tajwar	Rangpur	EEE	2024-11-25 16:00:00
Edit Copy Delete	4	Maruf	Hasan	Barisal	EEE	2024-11-25 17:00:00
Edit Copy Delete	5	Ashek	Farabi	Gazipur	TEX	2024-11-25 18:00:00
Edit Copy Delete	6	Ismile	Hasan	Barisal	TEX	2024-11-25 19:00:00

9.SELECT student.s_id, student.FirstName, student.LastName

FROM student

INNER JOIN course_registration ON student.s_id = course_registration.s_id

WHERE course_registration.s_id = 142002015;

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
SELECT student.s_id, student.FirstName, student.LastName FROM student INNER JOIN course_registration ON student.s_id =
course_registration.s_id WHERE course_registration.s_id = 142002015;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

s_id	FirstName	LastName

10. SELECT student.s_id, student.FirstName, student.dept_name, department.dept_id
FROM student
INNER JOIN department ON student.dept_name = department.dept_name

```
SELECT student.s_id, student.FirstName, student.dept_name, department.dept_id FROM student INNER JOIN department ON
student.dept_name = department.dept_name;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

s_id	FirstName	dept_name	dept_id
1	Zeseya	CSE	101
2	Sakib	CSE	101
3	Asef	EEE	102
4	Maruf	EEE	102
5	Ashek	TEX	103
6	Ismile	TEX	103

Console Bookmarks Opt

11. SELECT student.s_id, student.FirstName, student.dept_name, department.dept_id,
course_registration.course_code
FROM student
INNER JOIN department ON student.dept_name = department.dept_name
INNER JOIN course_registration ON department.dept_id = course_registration.dept_id;

12.SELECT student.s_id, student.FirstName, student.dept_name, department.dept_id, course_registration.course_code

FROM student

INNER JOIN department ON student.dept_name = department.dept_name

INNER JOIN course_registration ON department.dept_id = course_registration.dept_id

GROUP BY s_id;

6 Lab Exercise (Submit as a report)

oders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.50	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.50	2012-08-17	3009	5003
70007	948.50	2012-09-10	3005	5002
70005	2400.60	2012-07-27	3007	5001
70008	5760.00	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70002	2480.40	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.60	2012-04-25	3002	5001

customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3004	Fabian Johnson	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Cameron	Berlin	100	5003
3008	Julian Green	London	300	5002
3001	Brad Guzan	London		5005
3003	Jozy Altidore	Moscow	200	5007

Figure 6: Customer and Salesman table

1. Write a SQL statement to find the details of a order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and commission rate he gets for an order.

customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3004	Fabian Johnson	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Cameron	Berlin	100	5003
3008	Julian Green	London	300	5002
3001	Brad Guzan	London		5005
3003	Jozy Altidore	Moscow	200	5007

salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hense		0.12
5007	Paul Adam	Rome	0.13

Figure 7: Customer and Salesman table

2. Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.
3. Attach with query codes and with output screenshots in the report.

ANSWER TO THE Q NO: 01

1. Write a SQL statement to find the details of an order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and how much commission he gets for an order.

```
SELECT o.ord_no,
    o.ord_date,
    o.purch_amt,
    c.cust_name AS "Customer Name",
    s.name AS "Salesman",
    s.commission
 FROM orders o
```

```
INNER JOIN customer c
  ON o.customer_id=c.customer_id
INNER JOIN salesman s
  ON o.salesman_id=s.salesman_id;
```

ANS TO THE Q NO: 02

2. Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.

```
SELECT c.cust_name AS "Customer Name"
  FROM customer c
  LEFT JOIN salesman s
    ON c.salesman_id=s.salesman_id
  ORDER BY c.customer_id ASC;
```