



# SILVER OAK UNIVERSITY

---

EDUCATION TO INNOVATION

## CERTIFICATE

*This is to certify that*

Mr. /Ms PANCHAL YASH ANIL

With enrollment no 240240305000247 from

Semester 1<sup>st</sup> Div. B has Successfully

completed his/her laboratory experiments in the **Database Management System Principles (4040773101)** from the

department of

BACHELOR OF COMPUTER SCIENCE & IT during the

academic year 2024 - 25.

Date of Submission: \_\_\_\_\_

Staff In charge: *Shyadore*

Head of Department



# SILVER OAK UNIVERSITY

---

## EDUCATION TO INNOVATION

### EXPERIMENT NO: 1

**Aim:** Create a database and a few tables with basic SQL commands

- a) CREATE DATABASE
- b) USE DATABASE
- c) CREATE TABLE

**Description:** The CREATE DATABASE statement is used in SQL to create a new database. It specifies the name of the database that you want to create.

The USE DATABASE statement is used to select a specific database for use in subsequent SQL queries and commands. Once a database is selected using USE, all subsequent queries will operate within that database context until a different database is selected or the connection is terminated.

The CREATE TABLE statement is used to create a new table within a selected database. Tables in SQL databases are used to store structured data, where each table consists of rows (records) and columns (fields) that define the data structure.

**Procedure:**

```
CREATE TABLE STUDENTS  
STUDENTS ID INT PRIMARY KEY,  
Name VARCHAR (50),  
Age INT,  
Course VARCHAR (50);  
INSERT INTO Students (Student ID, Name, Age, Course)  
Values (1, 'Yash Panchal', 20, 'Computer Science');  
(2, 'Devashish Panchal', 21, 'Mechanical Engineering');  
(3, 'Hetal Panchal', 22, 'MBA');  
(4, 'Pankaj Panchal', 23, 'Business Management');  
INSERT INTO STUDENTS (Student ID, Name, Age, Course)  
Select Employee ID, Employee Name, Employee Age  
'General Studies' FROM Employees WHERE  
Department = 'Training';  
SELECT * FROM Student
```



# SILVER OAK UNIVERSITY

---

## EDUCATION TO INNOVATION

EmployeeID	lastname	firstname	Department	Date
(123)	(VAR)	(VAR)	(VAR)	Date

### Conclusion:

This Program exemplifies essential database operations reinforcing the importance of database in managing structure duties effectively

### References:

- [1] <https://www.geeksforgeeks.org/sql-create-database/>

### Post Practical Question:

1. What SQL command is used to create a new database?

Ans- CREATE DATABASE

2. What SQL command is used to select a database for use in subsequent queries?

Ans- USE

3. Explain the purpose of the PRIMARY KEY constraint in SQL tables.

Ans- THE PRIMARY KEY CONSTRAINT IN SQL TABLE PLAYS A CRUCIAL ROLE IN MAINTAINING THE INTEGRITY AND STRUCTURE OF RELATIONAL DATABASE

Signature with Date of Completion	
Marks out of 10	



# SILVER OAK UNIVERSITY

---

## EDUCATION TO INNOVATION

### EXPERIMENT NO: 2

**Aim:** Implement SQL queries to perform various DDL Commands.

a) Create a minimum 5 tables with different data types and operate upon them.

**Description:**

To implement SQL queries for Data Definition Language (DDL) commands, you can create tables with various data types and perform operations such as altering tables, dropping tables, and modifying table structures. Below is an example of creating a set of tables with different data types and performing basic operations on them

**Procedure:**

```
CREATE TABLE Students C
Student ID PRIMARY KEY .
FIRST NAME VARCHAR(50)
LAST NAME VARCHAR (50),
AGE INT
ENROLMENT DATE);

CREATE TABLE COURSES C
COURSE ID INT PRIMARY KEY,
COURSE NAME VARCHAR (100),
CREDIT INT);

CREATE TABLE ENROLLEMENT
ENROLMENT ID INT PRIMARY KEY
STUDENT ID INT,
COURSE ID INT,
ENROLMENT DATE DATE,
FOREIGN KEY (STUDENTID)
REFERENCES STUDENTS (STUDENT-ID)
FOREIGN KEY (COURSE ID)
REFERENCES COURSE (COURSE ID);
```



# SILVER OAK UNIVERSITY

---

## EDUCATION TO INNOVATION

~~ALTER TABLE STUDENTS~~

~~ADD GENDER CHAR(1);~~

~~ALTER TABLE COURSES~~

~~ADD DEPARTMENT VARCHAR(50);~~

~~DROP TABLE COURSES;~~

~~DROP TABLE STUDENTS;~~

*Customer Table*

CustomerID	Name	Email	Address
1	John	xyz@gmail	New York
2	Jane	mn@gmail	London

*Product Table*

ProductID	ProductName	Price
1	Phone	500.00
2	Laptop	100000

Conclusion:

The implementations of various data definition language (DDL) Commands in the SQL Program provide essential insights into the structure and management of a relational database.

*Orders Table*

order	CustomerID	Date
1	1	2024-1-22
2	2	2024-1-24

*Departments Table*

DepartmentID	Department
1	Sales
2	Marketing



# SILVER OAK UNIVERSITY

---

## EDUCATION TO INNOVATION

### EXPERIMENT NO: 3

**Aim:** Implement SQL queries to perform various DML Commands.

- Insert minimum 10 rows using different insert methods
- Edit and Remove data using update and delete commands.

**Description:** Implementing SQL queries for Data Manipulation Language (DML) commands involves inserting, updating, and deleting data within a database. These commands are essential for modifying existing data and ensuring the accuracy and integrity of database content.

**Procedure:**

```
CREATE DATABASE COMPANY ;
USE COMPANY ;
CREATE TABLE EMPLOYEES (
EMPLOYEES ID INT PRIMARY KEY ,
FIRST NAME EMPLOYEES (
EMPLOYEES ID INT PRIMARY KEY ,
First Name VARCHAR (50),
Last Name VARCHAR (50),
Hire Date,
Salary INT;
```

```
INSERT INTO EMPLOYEES (EMPLOYEEID, FIRSTNAME, HIREDATE, SALARY) VALUES ('1', 'Emily', 'Davis', '2020-07-15', 80000.00);
('2', 'John', 'Doe', '2020-01-15', 6000.00),
('3', 'Mike', 'Johnson', '2021-06-30', 50000.00),
('4', 'Bob', 'Crane', '2019-12-15', 62000.00),
('5', 'Devarsh Panchal', '2020-10-07', 50,000.00),
('6', 'Sophie', 'Crane', '2023-09-01', 62200.00),
('7', 'Mani', 'Sinha', '2019-01-11', 4000.00),
('8', 'Tom', 'Patel', '2020-03-12', 50000.00),
('9', 'Bob', 'Doe', '2019-05-05', 38000.00),
('10', 'Maheshi Sinha', '2017-11-29', 3200.00)
```

**Conclusion:**

The DML Program Effectively illustrates how to manage data within a relational database. The skills demonstrated are essential for any data-driven application as everything the data remains accurate relevant and easily accessible. This foundation knowledge



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

help to enables further exploration  
to more complex SQL  
and data functionalities

Customer ID	Name	Email
1	John D	JohnD@Gmail.com
2	Bob	Bob@gmail.com
3	Alice John	Alice@gmail.com
4		

### Post Practical Question:

1. Describe at least three different methods you used to insert data into the Students table in your SQL program.

~~Ans-BASIC INSERT STATEMENT :- This is the simplest method to add a single row into the table~~  
~~insert multiple rows in one statement or you can insert multiple rows at once by listing multiple set of values.~~

2. How would you update the Age of the student named Linda Davis' in the Students table to 22 years old?

~~Ans- UPDATE Student  
Set age = 22,~~

~~Where name = "Entity Davis";~~



# SILVER OAK UNIVERSITY

DATA SCIENCE

## EDUCATION TO INNOVATION

### EXPERIMENT NO: 4

Ques: Retrieve data using SQL commands and various SQL operators.

**Description:** In SQL (Structured Query Language), the **SELECT** statement is used to retrieve data from a database. It allows you to specify the columns you want to retrieve data from and can be combined with various operators to filter, sort, and manipulate the data before presenting it.

#### Principle

SQL

~~SELECT 1 Column 1 , column 2  
FROM table - Name;~~

SQL

~~SELECT \* FROM customers;  
SELECT column 1 , column 2 ...  
FROM table - Name;  
WHERE Condition;~~

SQL

~~SELECT \* FROM customers;  
SELECT column 1 , column 2 ...  
FROM table - Name;  
WHERE Condition;~~

SQL

~~SELECT column 1 , column 2 ...  
FROM table - Name;  
WHERE Condition;  
GROUP BY column 1 ;  
HAVING Condition;~~

SQL

~~SELECT column 1 , column 2 ...  
FROM table - Name;  
WHERE Condition;  
GROUP BY column 1 ;  
HAVING Condition;~~

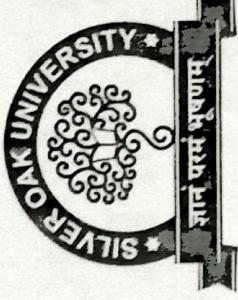
#### Conclusion

Additional SQL Operators  
 → Arithmetic Operators +,-, \*, /, %  
 → Comparison Operators =, <, >, <=, >=, !=  
 → Logical Operators AND, OR, NOT  
 → String Operators LIKE, IN, BETWEEN  
 By executing using these operators and the **SELECT Statement**, you can perform powerful data retrieval and analysis based on SQL.

ID	Name	Age
1	John	25
2	Jane	30

ID	Name	Age
1	John	25

ID	Name	Age
1	John	25



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

## EXPERIMENT NO: 5

Aim: Implement SQL queries using Date functions like add-

- a) months.
- b) months-between
- c) round
- d) next-day
- e) truncate
- f) greatest
- g) new-time

Description: Below is a short description of each Date function along with an example SQL code snippet for its implementation

- a) months - Adds a specified number of months to a date value
- b) months-between - Calculates the number of months between two dates
- c) round - Rounds a date to the nearest unit specified (day, month, year, etc.)
- d) next-day - Finds the next specific day of the week after a given date.
- e) truncate - Truncates a date to a specified unit (day, month, year, etc.).
- f) greatest - Returns the greatest value among the list of date expressions.
- g) new-time - Converts a date and time from one time zone to another.

Procedure:

### Add Month to a Date

```
SELECT ADD_MONTHS(CORDER_DATE,3) AS three_months_later  
FROM orders;
```

### 2) Calculating months Between Two Dates

```
SELECT MONTHS_BETWEEN(CSYSDATE, CORDER_DATE) AS months_since_order  
FROM orders;
```

### 3) Rounding a Date to the Nearest Month

```
SELECT ROUND(Corder_date,'Month') AS rounded_date  
FROM orders;
```

### 4) Finding the Next Monday after a date :

```
SELECT NEXT_DAY(Corder_date,'Monday') AS next-monday  
FROM orders;
```



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

#### Conclusion:

Truncating a date to the start of the month

```
Select TRUNC(Order_date, 'month')  
AS start_of_month FROM orders;  
FINDING THE LATEST ORDER STATEMENT  
MULIPLE ORDERS
```

Select (reatest Order\_date) dated  
order\_date ) AS latest\_order;  
date FROM orders ;

The exact syntax and available functions might vary slightly depending on the specific SQL database system. Leaving your database documentation, consult the documentation for your specific database system to learn about additional date and time functions.

#### Post Practical Question:

- What does the ADD\_MONTHS function in SQL do?
  - Adds a specified number of days to a date
  - Adds a specified number of months to a date
  - Adds a specified number of years to a date
  - Adds a specified number of hours to a date

Answer **Adds a specified number of months**

- Which SQL function calculates the number of months between two dates?
  - MONTH DIFFERENCE
  - MONTHS BETWEEN
  - MONTH DIFF
  - MONTHS DIFFERENCE

Answer **MONTHS BETWEEN**

- What does the ROUND function in SQL do when used with dates?
  - Rounds the date to the nearest hour
  - Rounds the date to the nearest day
  - Rounds the date to the nearest month
  - Rounds the date to the nearest year

Answer **rounds the date to the nearest day**

- Which SQL function finds the next specified day of the week after a given date?
  - NEXT DAY
  - FIND NEXT DAY
  - NEXT DAY



# SILVER OAK UNIVERSITY

---

EDUCATION TO INNOVATION

D. FIND DAY

Answer:

ANSWER - DAY

5. When does the TREC Committee meet each month?

- A. From 1st to the 10th minute
- B. From 1st to the 2nd hour
- C. From 1st to the 10th day
- D. From 1st to the 10th second

Answer: TREC meets the 1st to the 10th minute.

RESULT	2022-01-01
--------	------------

RESULT	3
--------	---

RESULT	2022-02-01
--------	------------

RESULT	2022-01-03
--------	------------

Signature with Date of Completion	<i>Abdullah</i>
Marks out of 10	



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

## EXPERIMENT NO: 6

**Aim:** Implement SQL queries using Numeric functions like

- a) Abs
- b) Ceil
- c) Exp
- d) Floor
- e) Power
- f) Mod
- g) Round
- h) Trim
- i) sqrt

**Description:**

- a) Abs - Returns the absolute value of a numeric expression.
- b) Ceil - Returns the smallest integer greater than or equal to a specified number.
- c) Exp - Returns the value of e raised to the power of a specified number.
- d) Floor - Returns the largest integer less than or equal to a specified number.
- e) Power - Returns a number raised to the power of another number.
- f) Mod - Returns the remainder of a division operation.
- g) Round - Rounds a number to a specified number of decimal places.
- h) Trim - Removes specific characters from the beginning or end of a string.
- i) sqrt - Returns the square root of a non-negative number.

**Procedure:** A) Abs

SELECT ABS(50) AS result;

b) Ceil

SELECT CEIL(10.5) AS result;

c) Exp

SELECT EXP(2) AS result;

d) Floor

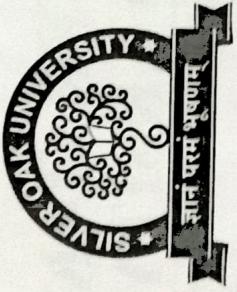
SELECT FLOOR(10.5) AS result;

e) Power

SELECT POWER(2,3) AS result;

f) Mod

SELECT MOD(10,3) AS result;



# SILVER OAK UNIVERSITY

---

EDUCATION TO INNOVATION

## ⇒ Round

SELECT ROUND(10.567, 2) AS result;

SELECT TRIM('HelloWorld') AS result;  
" " Scan

SELECT SQRT(16) AS result;

## Conclusion:

abs	ceil	exp	floor
20	1!	1.386e+308	10

mod	round	sqrt
4	10.567,2	4

power	result	trim
2	10.567,2	HelloWorld

## Post Practical Question:

1. What does the ABS function in SQL do?
  - A. Returns the ceiling value of a number
  - B. Returns the floor value of a number
  - C. Returns the absolute value of a number
  - D. Returns the square root of a number

Answer C) Returns the absolute value of a number



# SILVER OAK UNIVERSITY

---

EDUCATION TO INNOVATION

## EXPERIMENT NO: 7

AIM: Implement SQL queries using Character functions like in it

- a) ltrim
- b) lcoer
- c) upper
- d) RTRIM
- e) trim
- f) translate
- g) replace
- h) substring

### Description:

- a) ltrim - Converts the first character of a string to uppercase and all other characters to lowercase.
- b) lower - Converts all characters in a string to lowercase.
- c) upper - Converts all characters in a string to uppercase.
- d) lcoer - Removes leading spaces (or specified characters) from a string.
- e) trim - Removes trailing spaces (or specified characters) from a string.
- f) translate - Translates characters in a string from one set to another set of characters.
- g) replace - Replaces occurrences of a specified substring with another substring in a string.
- h) substring - Extracts a substring from a string.

### Procedure: OR CAP

SELECT CAP ('HelloWorld') AS result;  
by lcoer

SELECT LOWER ('HelloWorld') AS result;  
by upper

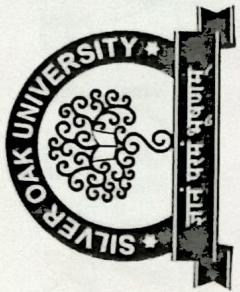
SELECT UPPER ('HelloWorld') AS result;  
by RTRIM

SELECT LTRIM ('HelloWorld') AS result;  
by ltrim

SELECT TRANSLATE ('HelloWorld', 'h', 'H')  
by translate

SELECT TRANSLATE ('HelloWorld', 'h', 'H')  
by translate

SELECT TRANSLATE ('HelloWorld', 'h', 'H')  
by translate



# SILVER OAK UNIVERSITY

---

EDUCATION TO INNOVATION

Q) Replace

```
SELECT REPLACE ('helloworld', 'world', 'earth')
AS result;
```

10) Substring

```
SELECT SUBSTRING ('helloworld' FROM 7)
AS result
```

Conclusion:

Q) Cap

result
hello world

Q) Upper

result
HELLOWORLD

Q) Trim

result
helloworld

Q) Translate

result
resukt

Q) Replace

result
hellocord

Q) Substring

result
world

Q) Left

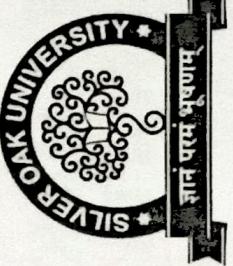
result
hello

Q) Right

result
world

### Post Practical Question:

- What does the INITCAP function in SQL do?
  - Converts all characters to uppercase.
  - Converts all characters to lowercase.
  - Converts the first character of each word to uppercase and the rest to lowercase.
  - Removes leading and trailing spaces from a string.
- Answer: C) Converts the **first character of each word to uppercase**, and the **rest to lowercase**.
  - SQL function converts all characters in a string to lowercase.
  - Which SQL function converts all characters in a string to lowercase?



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

## EXPERIMENT NO: 8

Aim: Implement SQL queries using Group functions like

a) Avg      b) Min      c) Max      d) Sum      e) Count

Description:

1. a) Avg - Calculates the average value of a set of values.
2. b) Min - Retrieves the minimum value from a set of values.
3. c) Max - Retrieves the maximum value from a set of values.
4. d) Sum - Calculates the sum of a set of values.
5. e) Count - Counts the number of rows returned by a query.

Procedure:

Q1 avg

SELECT AVG (Salary) FROM employee;

by min

SELECT MIN (Salary) FROM employee;

by max

SELECT MAX (Salary) FROM employee;

by sum

SELECT SUM (Salary) FROM employee;

by count

SELECT COUNT (\*) FROM employee;

Conclusion:

Q1 avg

by min

by max

by sum

by count

avg (Salary)	50,000	min (Salary)	20,000	max (Salary)	80,000	sum (Salary)	30,000
Count (*)				5			

avg (Salary)	50,000	min (Salary)	20,000	max (Salary)	80,000	sum (Salary)	30,000
Count (*)				5			



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

## EXPERIMENT NO: 9

Aim: Implement SQL queries using

- a) Group by
- b) Having
- c) Order by Clause

Description:

- a) Group by - The GROUP BY clause is used to group rows that have the same values under specific columns.
- b) Having - The HAVING clause is used to filter group results based on specified conditions, after the GROUP BY clause.
- c) Order by Clause - The ORDER BY clause is used to sort the result set by one or more columns.

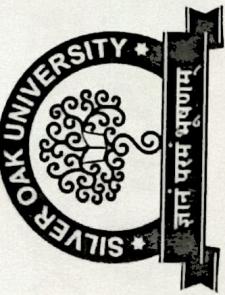
Procedure:

### By Group by

SELECT department , Avg(salary)  
FROM employees  
GROUP BY department ;  
b) Having

SELECT department , Avg(salary)  
FROM employees ;  
GROUP BY department  
Having Avg(salary) > 5000 ;

c) Order by clause  
SELECT \*  
FROM employees  
ORDER BY salary DESC ;



# SILVER OAK UNIVERSITY

---

EDUCATION TO INNOVATION

### Conclusion:

A) GROUP BY

Department	avg(Sales)
Sales	60,000
Marketing	55,000
IT	70,000

b) HAVING

Department	avg(Sales)
Sales	60000
IT	70000
Sales	60000

Role	appoinment	Salary
man	IT	80,000
john	IT	70,000
mike	Sales	70,000
bob	marketing	60,000

### Post Practical Question:

1. What is the primary purpose of the GROUP BY clause in SQL?
  - A. To filter rows based on specified conditions.
  - B. To sort the result set in ascending order.
  - C. To group rows that have the same values into summary rows.
  - D. To join tables based on common columns.
2. When is the HAVING clause used in SQL queries?
  - A. Before the GROUP BY clause to filter rows.
  - B. After the ORDER BY clause to filter rows.
  - C. After the GROUP BY clause to filter groups based on conditions.
  - D. Before the ORDER BY clause to filter rows.
3. Which SQL clause is used to sort the result set in a specified order?
  - A. GROUP BY
  - B. HAVING
  - C. ORDER BY
  - D. WHERE
4. What does the ORDER BY clause in SQL allow you to do?

Answer: ~~After the GROUP BY clause to filter groups based on conditions.~~  
~~After the GROUP BY clause to filter rows.~~  
~~Before the ORDER BY clause to filter rows.~~  
~~Before the ORDER BY clause to filter rows.~~

Answer: ~~After the GROUP BY clause to filter rows.~~  
~~After the GROUP BY clause to filter groups based on conditions.~~  
~~Before the ORDER BY clause to filter rows.~~  
~~Before the ORDER BY clause to filter rows.~~

Answer: ~~After the GROUP BY clause to filter rows.~~  
~~After the GROUP BY clause to filter groups based on conditions.~~  
~~Before the ORDER BY clause to filter rows.~~  
~~Before the ORDER BY clause to filter rows.~~



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

## EXPERIMENT NO: 10

Ans: Implement SQL queries using Set operators like

- a) Union
- b) union all
- c) intersect
- d) Minus

Description:

a) Union - The UNION operator is used to combine the result sets of two or more SELECT statements. It removes duplicate rows between the various SELECT statements.

b) union all - The UNION ALL operator is similar to UNION, but it does not remove duplicate rows between the SELECT statements. It returns all rows from all SELECT statements.

c) intersect - The INTERSECT operator returns only the rows that are common to the result sets of two SELECT statements. It effectively performs an intersection of the result sets.

d) Minus - The MINUS operator returns only the rows that appear in the first SELECT statement but not in the second SELECT statement. It effectively performs a set difference operation.

Procedure:

Q1) Union

~~SELECT \* FROM employees WHERE  
department = 'Sales'~~

~~UNION~~

~~SELECT \* FROM employees WHERE  
department = 'Marketing'~~

~~by Union All~~

~~SELECT \* FROM employees WHERE  
department = 'Sales'~~

~~Union All~~

~~SELECT \* FROM employees WHERE  
department = 'Marketing'~~



# SILVER OAK UNIVERSITY

---

EDUCATION TO INNOVATION

## CJ Interest

```

SELECT FROM employees WHERE
department = 'Sales'
INTERSECT
SELECT * FROM employees WHERE
Sales > 50000
    
```

↳ Minus

```

SELECT * FROM employees
WHERE department
= 'Sales'
SELECT * FROM employees
WHERE
Salary > 70000
    
```

↳ Minus

↳ Minus

↳ Minus

## Conclusion:

## q) union all

↳ Intersect

name	department	Salary
John	Sales	60000
Micheal	Marketing	55000
Bob	Sales	50000
John	Sales	60000

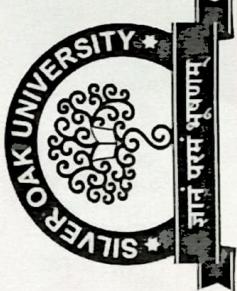
name	Department	Salary
John	Sales	60000
April	Marketing	55000
Bob	Sales	50000
John	Sales	60000

↳ Minus

name	Department	Salary
Bob	Sales	50000

- Post Practical Question:
- What does the UNION SQL operator do?
    - Returns only common rows between two SELECT statements.
    - Combines and returns all rows from multiple SELECT statements, including duplicates.
    - Combines and returns unique rows from multiple SELECT statements.
    - Returns rows from the first SELECT statement that are not present in the second SELECT statement.

Answer: Combines and returns unique rows from multiple SELECT statements



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

## EXPERIMENT NO: 11

**Aim:** Retrieve data spread across various tables or the same table using various Joins.

- a) Left join
- b) Right join

**Description:**

- a) Left join - The LEFT JOIN (or LEFT OUTER JOIN) retrieves all records from the left table (table1), and the matched records from the right table (table2). If there is no match, NULL values are returned for the right table columns.
- b) Right join - The RIGHT JOIN (or RIGHT OUTER JOIN) retrieves all records from the right table (table2), and the matched records from the left table (table1). If there is no match, NULL values are returned for the left table columns.

**Procedure:**

by left join

~~SELECT \*~~  
~~FROM employees~~  
~~LEFT JOIN Departments~~  
~~ON employees .dept =~~  
~~Departments . dept~~  
by Right join

~~SELECT \*~~  
~~FROM employees~~  
~~RIGHT JOIN departments~~  
~~ON employee . dept =~~  
~~departments . dept~~



# SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Conclusion:

Q) Left join

Employee ID	Name	Address	Department
1	John	101	Sales
2	Alice	102	Marketing
3	Bob	101	IT
4	Mike	101	IT

b) Right Join

ID	Name	Dept	Classification
1	John	101	sales
3	Bob	102	Marketing
2	Hab	102	vacant
4	Tom	103	IT

Post Practical Question:

1. What does a LEFT JOIN in SQL do?  
  - A. Retrieves all records from both tables.
  - B. Retrieves all records from the left table and matching records from the right table.
  - C. Retrieves all records from the right table and matching records from the left table.
  - D. Retrieves only matching records from both tables.

Answer: B. Retrieves all records from the left table and matching records from the right table.
2. When using a LEFT JOIN, what happens if there are no matching rows in the right table?  
  - A. All rows from both tables are included.
  - B. All rows from the left table are included with NULL values for the right table columns.
  - C. All rows from the right table are included with NULL values for the left table columns.
  - D. Only matching rows from both tables are included.

Answer: B. All rows from the left table are included with NULL values for the right table columns.
3. Which SQL join retrieves all records from the right table and matching records from the left table?  
  - A. INNER JOIN
  - B. LEFT JOIN
  - C. RIGHT JOIN
  - D. FULL OUTER JOIN

Answer: C. RIGHT JOIN
4. What does a RIGHT JOIN in SQL do?  
  - A. Retrieves all records from both tables.
  - B. Retrieves all records from the left table and matching records from the right table.
  - C. Retrieves all records from the right table and matching records from the left table.