Lab Manual

of

Database Management Systems

Subject Code: CS 313 Class: ICD Vth Semester

Prepared by:

Dr. Jagdeep Singh

Assistant Professor (CSE)



Department of Computer Science and Engineering

Sant Longowal Institute of Engineering and Technology, Longowal, India

October 2024

CERTIFICATE

This is to certify that this manual is a bonafide record of practical work in the **DBMS Lab** in **V**th **Semester of III Year ICD (CSE) program** during the academic year **2024-25**. This content was prepared by **Dr. Jagdeep Singh (Assistant Professor),** Department of Computer Science and Engineering.

Dr. Jagdeep Singh AP, CSE, SLIET LONGOWAL

Syllabus

Title of the course : Database Management Lab

Subject Code : CS-313

Course outcomes: At the end of the course, students will be able to:

CO1	Devise queries using DDL, DML, DCL and TCL commands.
CO2	Develop application programs using PL/SQL.
CO3	Create views, forms and reports.
CO4	Familiarization with different types of keys.

CO/PO M	apping : (Strong(S)	/Medium(M)/Weak	(W) indic	ates stren	gth of cor	relation)		
COs				P	rogramme	Outcome	es (POs)			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		S	S	М						S
CO2		S	М	S						S
CO3		S	S	М						S
CO4		S	М	М						S

LIST OF PRACTICALS

- 1. Introduction to various constraints such as Primary key, secondary key, Super key, etc.
- 2. To study Data Definition Commands (create, drop).
- 3. To study Data Manipulation Commands (insert, delete, update, select)
- 4. To study Data Control Commands (Commit, revoke, rollback, connect, execute)
- 5. Create Table, SQL for Insertion, Deletion, Update and Retrieval using aggregating functions.
- 6. Write Programs in PL/SQL, Understanding the concept of Cursors.
- 7. Write Program for Join, Union & intersection etc.
- 8. Creating Views, Writing Assertions, and Triggers.
- 9. Creating Forms, Reports etc.
- 10. WAP in PL/SQL for adding two numbers.
- 11. WAP in PL/SQL for reversing the number. For example the number is 12345 and reverse number will be 54321.
- 12. WAP in PL/SQL to find the number is even or odd.
- 13. WAP in PL/SQL to count numbers from 1 to 100.

PREFACE

This "**DBMS**" lab manual is designed to teach the fundamentals of data base design, emphasizing implementation in the SQL/PLSQL programming language. Readers of this manual are expected to be familiar with the syntax of SQL and similar procedural languages. DBMS concepts are increasingly critical to the IT industry, particularly for software development at the system level.

This practical manual has been carefully prepared to enhance the development of procedural programming skills. It includes a variety of exercises and their solutions so that students can understand them quickly and easily. This manual will prove valuable to Computer Science & Engineering students in grasping the applied aspects of database. There is always room for improvement, and we welcome suggestions from readers and users for future editions.

BY Dr. Jagdeep Singh AP, CSE, SLIET LONGOWAL

ACKNOWLEDGEMENT

It was a wonderful experience working on the "DBMS Lab" manual. First, I would like to express my sincere gratitude to **Prof. Birmohan Singh**, Head of the Department of Computer Science and Engineering, for his continuous support and technical guidance in preparing this document. I am deeply indebted and would like to acknowledge the invaluable support and patronage of **Prof. Mani Kant Paswan**, Director of the institute, for providing me with this excellent opportunity and his constant encouragement throughout the process. Finally, I extend my heartfelt thanks to the entire faculty of the CSE Department, whose inspiration and assistance helped me achieve this goal.

BY Dr. Jagdeep Singh AP, CSE, SLIET LONGOWAL

General Instructions

- 1. **Punctuality**: Students must arrive on time for the DBMS lab. Latecomers will not be allowed to participate in the lab session.
- 2. Attendance: Programs missed due to tardiness will be avoided. Students are expected to be on time.
- 3. **Preparation**: Students should prepare at home for the sessions' scheduled Programs.
- 4. **ID Cards**: Displaying an identity card is mandatory for entry into the lab.
- 5. Mobile Phones: Students need help to bring mobile phones into the lab.
- 6. **Responsibility for Equipment**: Any damage or loss of equipment, such as keyboards or mice, during the lab session will be the student's responsibility. A penalty or fine will be imposed if necessary.
- 7. Lab Records: Students must update their lab observation books and records after each session. Before leaving, they must get their lab observation book signed by the faculty member.
- 8. **Submission of Lab Records**: Lab records must be submitted to the faculty in the staffroom during the next lab session for correction and return.
- 9. Movement in the Lab: Students should remain at their assigned stations and avoid moving around during the lab session.
- 10. Emergencies: In an emergency, students must obtain written permission from the faculty member in charge.
- 11. **Disciplinary Action**: Faculty members can suspend students from the lab session for disciplinary reasons.
- 12. **Original Work**: Students should not copy outputs from others. They must write their own results.

BY

Dr. Jagdeep Singh

LIST OF PROGRAMS

- 1. Introduction to various constraints such as Primary key, secondary key, Super key, etc.
- 2. To study Data Definition Commands (create, drop).
- 3. To study Data Manipulation Commands (insert, delete, update, select)
- 4. To study Data Control Commands (Commit, revoke, rollback, connect, execute)
- 5. WAP in SQL for learning the concept VIEWS
- 6. WAP in SQL to understand the concept of MIN, MAX, AVG and COUNT, etc.
- 7. WAP in SQL using ORDER BY and GROUP BY clause.
- 8. WAP in SQL to understand the concept of subqueries.
- 9. WAP in SQL using FOREIGN KEY.
- 10. WAP in SQL to learn the concept of RANK() and DENSERANK().
- 11. WAP in SQL to CONCAT Function.
- 12. WAP in SQL to learn the concept for STRING FUNCTION.
- 13. WAP in SQL for CONDITIONAL statements
- 14. WAP in SQL to learn TRANSCATION.
- 15. WAP in SQL to learn the concept of TRIGGER.
- 16. WAP in SQL using UNION and INTERSECT
- 17. WAP in SQL for Join, Union & intersection etc.
- 18. WAP in PL/SQL for adding two numbers.
- 19. WAP in PL/SQL for reversing the number. For example the number is 12345 and reverse number will be 54321.
- 20. WAP in PL/SQL to find the number is even or odd.
- 21. WAP in PL/SQL to count numbers from 1 to 20.

Program 1. Introduction to various constraints such as Primary key, secondary key, Super key, etc.

1. Primary Key

- A unique identifier for each record in a table.
- Characteristics:
 - Must contain unique values.
 - Cannot contain NULL values.
 - There can only be one primary key per table, which may consist of a single or multiple columns (composite key).

2. Secondary Key

- An attribute or a set of attributes used to access records in a database but does not uniquely identify them.
- Characteristics:
 - Allows for efficient retrieval of records based on non-unique attributes.
 - Can contain duplicate values and NULLs.

3. Super Key

- A set of one or more attributes that can uniquely identify a record in a table.
- Characteristics:
 - Can include primary keys and any additional attributes.
 - Not necessarily minimal; a super key may contain extra attributes beyond those needed for uniqueness.

4. Candidate Key

- A minimal super key, meaning it uniquely identifies a record without any extra attributes.
- Characteristics:
 - A table can have multiple candidate keys.
 - \circ $\,$ One candidate key is chosen as the primary key.

5. Foreign Key

• An attribute or a set of attributes in one table that refers to the primary key in another table.

• Characteristics:

- Establishes a relationship between two tables.
- Can contain duplicate values and NULLs.

```
CREATE DATABASE Employees;
USE Employees;
-- Create the "departments" table
CREATE TABLE departments (
  dept_id INT PRIMARY KEY,
                                  -- Primary Key
  dept name VARCHAR(50) UNIQUE
                                       -- Unique constraint
);
-- Insert sample data into "departments"
INSERT INTO departments (dept id, dept name) VALUES (1, 'HR');
INSERT INTO departments (dept id, dept name) VALUES (2, 'Finance');
INSERT INTO departments (dept_id, dept_name) VALUES (3, 'IT');
-- Create the "employees" table
CREATE TABLE employees (
  emp id INT PRIMARY KEY, -- Primary Key
  emp name VARCHAR(50) NOT NULL, -- Not Null constraint
  dept id INT,
                         -- Foreign Key constraint
  emp email VARCHAR(100) UNIQUE, -- Unique constraint
  -- Define a foreign key constraint referencing "departments" table
  FOREIGN KEY (dept id) REFERENCES departments(dept id)
);
-- Insert sample data into "employees"
INSERT INTO employees (emp id, emp name, dept id, emp email)
VALUES (1, 'Alice', 1, 'alice@example.com');
INSERT INTO employees (emp id, emp name, dept id, emp email)
VALUES (2, 'Bob', 2, 'bob@example.com');
INSERT INTO employees (emp id, emp name, dept id, emp email)
```

VALUES (3, 'Charlie', 1, 'charlie@example.com'); INSERT INTO employees (emp_id, emp_name, dept_id, emp_email) VALUES (4, 'David', 3, 'david@example.com'); -- Select query to display all records in the "departments" table SELECT * FROM departments; -- Select query to display all records in the "employees" table SELECT * FROM employees; -- Select query to retrieve employee names along with their department names SELECT e.emp_name, e.emp_email, d.dept_name

FROM employees e

JOIN departments d ON e.dept_id = d.dept_id;

MySQL Workbench	tance MySQL80 ×		- 0 ×
	e Server Tools Scripting Help		Ø []
Navigator	SQL File 1" × books authors customers		
SCHEMAS 💖	🗀 🖬 デ 😥 👰 🔘 🔂 💿 💿 🐻 Limit to 1000 rows 🔹 🙀 🚿	1 1	
Q. Filter objects	1 • CREATE DATABASE Employees;		
anabi anabi anabi autors bools customers ordertems driministration Schemas Information No object selected	2 ● USE Employees; 3 Create the "departments" table 4 ● ○ CREATE TABLE departments (dept_d INT PRIMARY KEY, Primary Key dept_name VARCHAR(S0) UNIQUE Unique constraint 7); 8 Insert sample data into "departments" 1 Result Grid : ***********************************	Wrap Cell Content: I	Rend
	Image: Second	1 row(s) affected 3 row(s) returned 4 row(s) returned	Form Efform Apply Revert
Object Info Session			
Ready			

MySQL Workbench		- 0 ×
SLIET DBMS × Local ins	tance MySQL80 ×	
<u>File Edit View Query Databas</u>	e <u>S</u> erver <u>T</u> ools <u>S</u> cripting <u>H</u> elp	
0 5 0 0 0 0 0		Ø 🔲 🗖
Navigator	SQL File 1* × books authors customers	
SCHEMAS 🚸	🗀 🖬 🕖 🖉 🖉 💿 😡 💿 💿 🐻 Limit to 1000 rows 🔹 🧙 🛫 🔍 🐧 😨	
Q Filter objects	1 • CREATE DATABASE Employees;	
▶ 🗐 anshi	2 • USE Employees;	
 onlinebookstore Tables 	3 Create the "departments" table	1
authors	4 ● ○ CREATE TABLE departments (
▶ books	5 dept_id INT PRIMARY KEY, Primary Key	
customers orderitems	6 dept_name VARCHAR(50) UNIQUE Unique constraint	
orders	7 ();	
Administration Schemas	8	
Information	9 Insert sample data into "departments"	
	A A THEFT THE I I I I I I I I I I I I I I I I I I I	
No object selected	Result Grid 📗 🛟 Filter Rows: Edit: 📸 📷 Export/Import: 🏭 🐻 Wrap Cell Content: 🗵	
	emp id emp name dept id emp email	
	1 Alice 1 alice@example.com	Result Grid
	2 Bob 2 bob@example.com	
	3 Charlie 1 charlie@example.com 4 David 3 david@example.com	
		Form Editor
	departments 1 employees 2 × Result 3	Apply Revert
	Outout	
	Action Output	
		Duration / Fetch
	Time Action Message 15 16:33:50 INSERT INTO employees (emp_id, emp_name, dept_id, emp_email) VALUES (3, ' 1 row(s) affected	0.000 sec
	16 16:33:50 INSERT INTO employees (emp_id, emp_name, dept_id, emp_email) VALUES (4, ' 1 row(s) affected	0.000 sec
	17 15:33:56 SELECT ' FROM departments LIMIT 0, 1000 3 3row(s) returned	0.000 sec / 0.000 sec
	18 16:33:56 SELECT * FROM employees LIMIT 0, 1000 4 row(s) returned	0.000 sec / 0.000 sec
	19 16:33:56 SELECT e.emp_name, e.emp_email, d.dept_name FROM employees e JOIN depart 4 row(s) returned	0.000 sec / 0.000 sec
Object Info Session		0.000 000 1
Ready		1
	🔍 Search 🛛 🥠 🖬 📱 🚱 🎲 💽 🧬 🏣 🔯 🖏 🔷 🛆 😭 🐄 🕬 🖄	⊃ ^{16:35} . 🧖

MySQL Workbench			- 0 ×
SLIET DBMS × Local ins	tance MySQL80 ×		
File Edit View Query Databas	se <u>S</u> erver <u>T</u> ools <u>S</u> cripting <u>H</u> elp		
	3 60 Q 🐙		0
Navigator	SQL File 1* × books authors customers		
SCHEMAS 🚸	🗀 🖬 🖉 🖉 👰 🕐 💁 📀 💿 🐻 Limit to 1000 rows 🔹 🙀 🔍	1 9	
Q Filter objects	1 • CREATE DATABASE Employees;		
▶ 🗐 anshi	2 • USE Employees;		
▼	3 Create the "departments" table		1
authors	4 ● ⊖ CREATE TABLE departments (
books	5 dept_id INT PRIMARY KEY, Primary Key		
 customers orderitems 	6 dept_name VARCHAR(50) UNIQUE Unique constraint		
ordentems orders	7);		
Administration Schemas	8		
Information	9 Insert sample data into "departments"		
Information	As a success time to a 21 b t1 t b A consider 26 limits		
No object selected	Result Grid 🔢 🚯 Filter Rows: Export: 🎼 Wrap Cell Content: 🏗		
	emp_name emp_email dept_name		Result Grid
	Bob bob@example.com Finance Alice alice@example.com HR		
	Charlie charlie@example.com HR		
	David david@example.com IT		Form Editor
	departments 1 employees 2 Result 3 ×		Read Only
	Output		
	Action Output -		
	# Time Action	Message	Duration / Fetch
	 15 16:33:50 INSERT INTO employees (emp_id, emp_name, dept_id, emp_email) VALUES (3, ' 	1 row(s) affected	0.000 sec
	16 16:33:50 INSERT INTO employees (emp_id, emp_name, dept_id, emp_email) VALUES (4, '	1 row(s) affected	0.000 sec
	 17 16:33:56 SELECT * FROM departments LIMIT 0, 1000 	3 row(s) returned	0.000 sec / 0.000 sec
	18 16:33:56 SELECT * FROM employees LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
Object Info Session	 19 16:33:56 SELECT e.emp_name, e.emp_email, d.dept_name FROM employees e JOIN depart 	. 4 row(s) returned	0.000 sec / 0.000 sec
Ready			1
	🛛 Q Search 🛛 🥠 🖬 🖳 🖉 🦸 🦿	📜 🕎 🖷 🔹 ^ 🗠 🖓 ENG 😤 da) 🖆	16:35 25-10-2024 📮 🧖

Program 2. To study Data Definition Commands (create, drop).

CREATE Command: Used to create new database objects such as tables, views, indexes, or schemas.

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

HireDate DATE

);

DROP Command: Used to delete existing database objects like tables, views, or indexes.

DROP TABLE Employees;

MySQL Workbench			- 0 ×
New Connection ×			
	Server Tools Scripting Help		
			Ø — —
Navigator	SQL File 1* SQL File 3* ×		
SCHEMAS ®	💼 🖬 🐓 😥 💿 🛐 💿 💿 🐻 Limit to 1000 rows 🔹 📩 🕩 🥝	. 1 =	
9. Filter objects	9 HireDate DATE		
▶ 🗐 anshi	10);		
employees onlinebookstore	11		
Tables	12 Insert values into the Employees table		1
 authors books 	13 • INSERT INTO Employees (EmployeeID, FirstName, LastName, HireDay	:e)	
customers	14 VALUES		
orderitems	15 (1, 'John', 'Doe', '2022-01-15'),		1
Administration Schemas	<pre>16 (2, 'Jane', 'Smith', '2021-03-10'), 17 (3, 'Robert', 'Johnson', '2023-07-05');</pre>		
Information	17 (3, 'Robert', 'Johnson', '2023-07-05'); 18		
No object selected	10		
No object selected	Result Grid 🔢 🚸 Filter Rows: Edit: 🔏 誌 Export/Import: 🙀	Wrap Cell Content:	
	EmployeeID FirstName LastName HireDate		
	1 John Doe 2022-01-15		Result Grid
	2 Jane Smith 2021-03-10 3 Robert Johnson 2023-07-05		
	3 Robert Johnson 2023-07-05		
			\sim
	Employees 1 ×		Apply Revert
	Output		
	Action Output -		
	# Time Action	Message	Duration / Fetch
	3 19:08:47 USE Sliet_emp	0 row(s) affected	0.000 sec
	4 19:08:52 SELECT * FROM Employees LIMIT 0, 1000	Error Code: 1146. Table 'sliet_emp.employees' doesn't exist	0.000 sec
	5 19:08:58 CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, FirstName VA.	0 row(s) affected	0.015 sec
	6 19:09:04 INSERT INTO Employees (EmployeeID, FirstName, LastName, HireDate) VALUES	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec
Object Info Session	7 19:09:08 SELECT * FROM Employees LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
Query Completed			
→ 29°C Clear	📲 Q Search 🛛 🥠 🖬 🔛 🔛 🥵 🥵	📲 🍼 🖾 🛛 🔺 🗠 🎜 🕅 🖘 🕼	19:09
Crown			

Program 3. To study Data Manipulation Commands (insert, delete, update, select)

INSERT INTO Employees (EmployeeID, FirstName, LastName, HireDate)

VALUES (1, 'John', 'Doe', '2023-01-15');

DELETE Command: Removes existing records from a table.

DELETE FROM Employees

WHERE EmployeeID = 1;

UPDATE Command: Modifies existing records in a table.

UPDATE Employees

SET LastName = 'Smith'

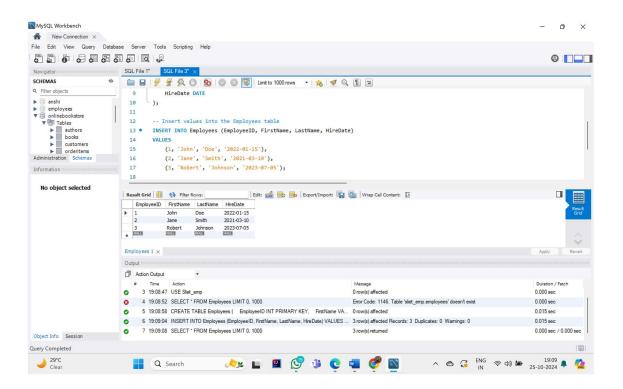
WHERE EmployeeID = 1;

SELECT Command: Retrieves data from one or more tables.

SELECT FirstName, LastName

FROM Employees

WHERE HireDate > '2022-01-01';



Programs 4

To study data control commands commit, revoke, rollback, connect, execute

COMMIT: Saves all changes made during the current transaction permanently in the database. Once committed, the changes cannot be rolled back. Syntax: COMMIT; **REVOKE:** Removes previously granted permissions or privileges from a user or role, which affects access to database objects like tables, views, or procedures. Syntax: REVOKE privilege type ON object name FROM user name; ROLLBACK: Reverts all changes made during the current transaction to the last committed state, undoing any modifications since the transaction began. Syntax: ROLLBACK; CONNECT: Establishes a connection between the user and the database, allowing them to perform operations within that session. Svntax: CONNECT user name/password@database name;

EXECUTE: Runs a specified SQL command or stored procedure, often used in procedural languages like PL/SQL to execute code blocks, queries, or stored functions.

Syntax: EXECUTE procedure_name;

Programs 5

WAP in SQL for learning the concept VIEWS

A view in SQL is a virtual table that is based on the result of a query. It does not store data itself; instead, it provides a way to present data from one or more tables in a specific format or structure.

Virtual Table: A view is treated like a table in SQL, but it does not hold any data. Instead, it generates data dynamically when queried.

Views are a powerful feature in SQL that enhance data management, security, and usability. They allow you to encapsulate complex queries and present data in a user-friendly manner, making them valuable tools for database design and interaction.

USE CompanyDB;

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Department VARCHAR(50),

Salary DECIMAL(10, 2));

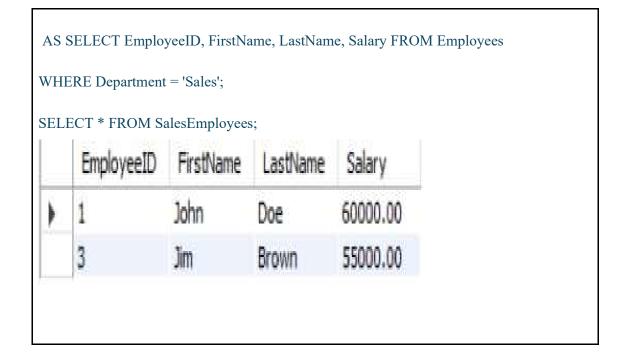
INSERT INTO Employees (EmployeeID, FirstName, LastName, Department, Salary) VALUES

- (1, 'John', 'Doe', 'Sales', 60000.00),
- (2, 'Jane', 'Smith', 'HR', 50000.00),
- (3, 'Jim', 'Brown', 'Sales', 55000.00),
- (4, 'Jake', 'White', 'IT', 70000.00);

SELECT * FROM Employees;

	EmployeeID	FirstName	LastName	Department	Salary
Þ	1	John	Doe	Sales	60000.00
	2	Jane	Smith	HR	50000.00
	3	Jim	Brown	Sales	55000.00
	4	Jake	White	Π	70000.00
	HULL	NULL	NULL	NULL	HULL

CREATE VIEW SalesEmployees



Programs 6

WAP in SQL to understand the concept of MIN,MAX,AVG and COUNT,etc.

The MIN function is used to retrieve the smallest value from a specified column in a table. It can be applied to numeric, date, or string columns.

The MAX function is used to find the largest value in a specified column. Similar to MIN, it can be applied to numeric, date, or string columns.

The AVG function calculates the average (mean) value of a numeric column. It sums all the values in the column and divides by the count of non-null entries.

The COUNT function counts the number of rows that meet a specified condition. It can be used to count all rows, non-null values in a column, or distinct values.

Create Database Employee; use Employee; create table employees(employee_id int primary key,

first_name varchar(20), last_name varchar(20), salary decimal(10,2), department varchar(10)

);

insert into employees(employee_id,first_name,last_name,salary,department) values

(1,"Priyanshu","Raj",90000,"Marketing"),

(2,"Sanskar","Raj",100000,"HR"),

(3,"Avinash","Aryan",900000,"Finance"),

(4,"Sanjeev","Kumar",900000,"IT"),

(5,"Krish","Rajan",100000,"Marketing"),

(6,"Aman","Raj",80000,"HR"),

(7,"Rahul","Raj",70000,"Finance"),

(8,"Alok","Raj",60000,"IT"); select * from

employees; select * from employees where

first_name like 'P%';

	employee_id	first_name	last_name	salary	department
•	1	Priyanshu	Raj	90000.00	Marketing
	2	Sanskar	Raj	100000.00	HR
	3	Avinash	Aryan	900000.00	Finance
	4	Sanjeev	Kumar	900000.00	п
	5	Krish	Rajan	100000.00	Marketing
	6	Aman	Raj	80000.00	HR
	7	Rahul	Raj	70000.00	Finance
	8	Alok	Raj	60000.00	IT

select * from employees where salary IS NULL;

select * from employees where salary IS NOT NULL;

select count(*) as total_employees from employees;

total_employees
 16

Select MAX(salary) as highest salary from employees;

highest_salary 900000.00

900000.00

Select MIN(salary) as lowest_salary from employees;

lowest_salary

▶ 10000.00

Select AVG(salary) as average_salary from employees;

ele	ct * from employe	1 - 2	1 -		
	employee_id	first_name	last_name	salary	departmen

Programs 7

WAP in SQL using ORDER BY and GROUP BY clause.

ORDER BY

Purpose: To sort the result set of a query based on one or more columns. Usage: It can sort data in ascending (ASC) or descending (DESC) order.

GROUP BY

Purpose: To group rows that have the same values in specified columns into summary rows, often used with aggregate functions (like COUNT, SUM, AVG).

Usage: It allows you to perform operations on each group of data.

CREATE DATABASE college;

USE College;

CREATE TABLE student(

roll_no INT PRIMARY KEY,

name VARCHAR(50),

marks FLOAT,

grade VARCHAR(5),

city VARCHAR(50)

);

INSERT INTO student(roll_no,name,marks,grade,city)

VALUES

(101,"Ram",97,"A","Delhi"),

(102,"Shyam",84,"B","Mumbai"),

(103,"Rohan",79,"C","Chennai"), (104,"Narayan",88,"B","Bangaluru"), (105,"Rakesh",63,"D","Kolkata"); SELECT DISTINCT grade FROM student; grade А ► в С D SELECT * FROM Student WHERE marks+3=100; roll_no name marks city grade 101 97 Delhi ۲ Ram A NULL NULL NULL NULL NULL ۰ SELECT * FROM Student WHERE marks>80 OR city="Kolkata"; roll_no name marks grade city ۲ 101 Ram 97 A Delhi в Mumbai 102 Shyam 84 Bangaluru 104 Narayan 88 В Rakesh D Kolkata 105 63 NULL NULL NULL NULL NULL . SELECT * FROM Student ORDER BY city ASC; roll_no name marks grade city 104 Narayan 88 В Bangaluru ۲ С 103 Rohan 79 Chennai Delhi 101 Ram 97 А 105 Rakesh 63 D Kolkata 102 Shyam 84 В Mumbai NULL NULL NULL NULL NULL ٠ SELECT * FROM Student ORDER BY marks DESC LIMIT 3;

	roll_no	name	marks	grade	city
•	101	Ram	97	Α	Delhi
	104	Narayan	88	В	Bangaluru
	102	Shyam	84	В	Mumbai
٠	NULL	NULL	NULL	NULL	NULL
SEL	ECT city,co	ount(name)			
FRO	M college.	Student			
Grou	up By city;				
	city	count(name)		
•	Delhi	1			
	Mumbai	1			
	Chennai	1			
	Bangaluru	1			
	Kolkata	1			
ant		4 1 X			
SEL	ECT city, a	vg(marks)			
FRO	M student				
GRC	OUP BY cit	V			
		-			
ORL	DER BY cit	-			
	city	avg(m	arks)		
•	Bangaluru	88			
	Chennai	79			
	Delhi	97			
	Kolkata	63			
	Mumbai	84			

Program 8

WAP in SQL to understand the concept of subqueries.

A subquery, or inner query, is a query nested inside another SQL query (the outer query). Subqueries can be used in various parts of a SQL statement, including the SELECT, FROM, and WHERE clauses. CREATE DATABASE retail store; USE retail_store; CREATE TABLE employees (employee id INT PRIMARY KEY, name VARCHAR(50), position VARCHAR(50), salary DECIMAL(10, 2)); CREATE TABLE customers (customer_id INT PRIMARY KEY, name VARCHAR(50), email VARCHAR(100)); INSERT INTO employees (employee id, name, position, salary) VALUES (1, 'Alice', 'Manager', 60000), (2, 'Bob', 'Sales', 50000), (3, 'Charlie', 'Sales', 50000), (4, 'David', 'Support', 40000); select*from employees; salary employee_id position name 1 Alice ۲ Manager 60000.00 2 Sales 50000.00 Bob Sale Sales 50000.00 3 Charlie 4 David Support 40000.00 NULL NULL NULL NULL

INSI	ERT INTO cust	omers (custome	er_id, name, email) VALUES
(1, 'J	ohn Doe', 'john	@example.com	ı'),
(2, 'J	ane Smith', 'jan	e@example.com	m');
selec	t*from custome	ers;	
	customer_id	name	email
•	1	John Doe	john@example. john@example.com
	2	Jane Smith	jane@example.com
	NULL	NULL	NULL
	8		
SEL	ECT name, sala	ry	
FRO	M employees		
WH	ERE salary > (S	ELECT AVG(salary) FROM employees);
	name	salary	
		60000.00	
	Alice	00000.00	
SEL	ECT name		
FRO	M employees		
WH	ERE employee_	id IN (SELEC	T customer_id FROM customers WHERE position = 'sales');
	name		
-			
►	Bob		

Program 9

WAP in SQL using FOREIGN KEY.

Referential integrity is a database concept that ensures relationships between tables remain consistent. It guarantees that a foreign key in one table must correspond to an existing primary key in another table. This helps maintain data accuracy and integrity across related tables.

```
Foreign Key Constraints: Referential integrity is enforced using foreign keys. A foreign key in a
child table points to a primary key in a parent table, ensuring that any value in the foreign key column
matches a value in the primary key column.
CREATE DATABASE School;
USE School:
-- Create Students table
CREATE TABLE Students (
  StudentID INT PRIMARY KEY,
  FirstName VARCHAR(50),
  LastName VARCHAR(50)
);
-- Create Courses table
CREATE TABLE Courses (
  CourseID INT PRIMARY KEY,
  CourseName VARCHAR(100)
);
-- Create Enrollments table with foreign keys
CREATE TABLE Enrollments (
  EnrollmentID INT PRIMARY KEY,
  StudentID INT,
  CourseID INT,
  FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
  FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
);
-- Insert sample data into Students
INSERT INTO Students (FirstName, LastName) VALUES ('John', 'Doe');
INSERT INTO Students (FirstName, LastName) VALUES ('Jane', 'Smith');
Select*from Students;
```

Inse INSE Select Inse INSE	RT INTO Courses; t*from Courses; CourseID 1 2 RULL ert sample data i	John Jane RULL nto Courses ees (CourseName) ees (CourseName)	VALUES ('Scien
Inse INSE Select Inse INSE	RT INTO Course RT INTO Course RT INTO Courses; t*from Courses; CourseID 1 2 NULL ert sample data i	nto Courses es (CourseName) es (CourseName) CourseName Mathematics Science	NULL VALUES ('Mathe VALUES ('Scien
Inse INSE Select Inse INSE	ert sample data i RT INTO Cours RT INTO Cours t*from Courses; CourseID 1 2 RULL ert sample data i	nto Courses ses (CourseName) ses (CourseName) CourseName Mathematics Science	VALUES ('Matho VALUES ('Scien
INSE INSE Select Inse INSE	RT INTO Cours RT INTO Cours t*from Courses; CourseID 1 2 RULL ert sample data i	es (CourseName) es (CourseName) CourseName Mathematics Science	VALUES ('Scien
INSE INSE Select Inse INSE	RT INTO Cours RT INTO Cours t*from Courses; CourseID 1 2 RULL ert sample data i	es (CourseName) es (CourseName) CourseName Mathematics Science	VALUES ('Scien
INSE Select Inse INSE	RT INTO Courses; t*from Courses; CourseID 1 2 RULL ert sample data i	CourseName Mathematics Science	VALUES ('Scien
Select	t*from Courses; CourseID 1 2 NULL ert sample data i	CourseName Mathematics Science	
 Inse INSE 	CourseID 1 2 NULL ert sample data i	CourseName Mathematics Science	
INSE] INSE]	1 2 NULL ert sample data i	Mathematics Science	
INSE] INSE]	2 NULL ert sample data i	Science	
INSE] INSE]	RULL ert sample data i	NULL	
INSE] INSE]	ert sample data i		
INSE] INSE]	-	nto Enrollments	
INSE] INSE]	-	nto Enrollments	
INSE	RT INTO Enrol		
		lments (StudentID	, CourseID) VAL
	RT INTO Enrol	lments (StudentID	, CourseID) VAL
SELE	ECT		
s.Fi	irstName,		
s.La	astName,		
c.C	CourseName		
FRON	M		
Enr	rollments e		
JOIN			
Stu	idents s ON e.Sti	udentID = s.Stude	ntID
JOIN			
		urseID = c.Course	-ID-
200	FirstName	LastName	CourseName
			Mathematics
		Doe	Science
•	John Jane	Smith	NCIEDCE

Programs 10.

WAP in SQL to learn the concept of RANK() and DENSERANK().

RANK: This function assigns a rank to each row within a partition of a result set. If two or more students have the same score, they will receive the same rank, and the next rank(s) will be skipped. For example, if two students are tied for rank 1, the next rank will be 3.

DENSE_RANK: Similar to RANK, but it does not skip ranks. If two students have the same score, they receive the same rank, but the next rank will be the immediate next number. For example, if two students are tied for rank 1, the next rank will be 2.

CREATE DATABASE Scho;

USE Scho;

CREATE TABLE Students (

StudentID INT PRIMARY KEY AUTO_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Score INT

```
);
```

INSERT INTO Students (FirstName, LastName, Score) VALUES

('Alice', 'Johnson', 85),

('Bob', 'Smith', 95),

('Charlie', 'Brown', 85),

('David', 'Wilson', 70),

('Eve', 'Davis', 95);

SELECT

StudentID,

FirstName,

LastName,

Score,

RAI	NK() OVER (ORDER BY S	core DESC) A	AS Ran		
FRC			,			
	tudents;					
	StudentID	FirstName	LastName	Score	Ran	
►	2	Bob	Smith	95	1	
	5	Eve	Davis	95	1	
	1	Alice	JOHNSON	85	3	
	3	Charlie	Brown	85	3	
	4	David	Wilson	70	5	
SEL	ECT					
S	tudentID,					
F	irstName,					
L	astName,					
	core,					
D	ENSE_RANK	K() OVER (OF	RDER BY Sco	ore DESC) AS DenseRa	ınk
FRC	DM					
S	tudents;					
	StudentID	FirstName	LastName	Score	DenseRank	
•	2	Bob	Smith	95	1	_
	5	Eve	Davis	95	1	
	1	Alice	Johnson	85	2	
	3	Charlie	Brown	85	2	
	4	David	Wilson	70	3	

Programs 11

WAP in SQL to CONCAT Function.

The **CONCAT** function in SQL is used to combine two or more strings into a single string. It is useful for creating full names, concatenating address fields, or forming any composite string from multiple sources.

CREATE DATABASE S;

USE S;

CREATE TABLE Students (

StudentID INT PRIMARY KEY AUTO_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50)

);

INSERT INTO Students (FirstName, LastName) VALUES

('Alice', 'Johnson'),

('Bob', 'Smith'),

('Charlie', 'Brown');

select *from Students;

	StudentID	FirstName	LastName
•	1	Alice	Johnson
	2	Bob	Smith
	3	Charlie	Brown
	NULL	NULL	NULL

SELECT

StudentID,

CONCAT(FirstName, '', LastName) AS FullName

FROM

Students;

	StudentID	FullName
•	1	Alice Johnson
	2	Bob Smith
	3	Charlie Brown

Program 12

WAP in SQL to learn the concept for STRING FUNCTION.

CRI	EATE DATABASE 1	raj;	
USE	E raj;		
CRI	EATE TABLE Stude	ents (
S	tudentID INT PRIM	ARY KEY AUTO I	NCREMENT,
F	irstName VARCHA	R(50),	
	astName VARCHAI		
);			
INS	ERT INTO Students	(FirstName, LastNa	me) VALUES
('Al	ice', 'Johnson'),		
('Bo	b', 'Smith'),		
('Ch	arlie', 'Brown'),		
('Da	wid', 'Wilson');		
	LECT		
	PPER(FirstName) A	S UpperFirstName	
	PPER(LastName) A	S UpperLastName	
FRO	OM		
S	tudents;		
	UpperFirstName	UpperLastName	
►	ALICE	JOHNSON	
	BOB	SMITH	
	CHARLIE	BROWN	
	DAVID	WILSON	
SEL	LECT		
L	OWER(FirstName)	AS LowerFirstName	,
L	OWER(LastName)	AS LowerLastName	
FRO	DM		
S	tudents;		
	,		

	LowerFirstNa	ame Lowe	erLastName		
•	alice	johns	on		
	bob	smith			
	charlie	brown	n		
	david	wilson	1		
SEL	ECT				
Fi	irstName,				
L	astName,				
L	ENGTH(FirstN	Name) AS Fi	rstNameLength,		
L	ENGTH(LastN	Jame) AS La	stNameLength		
FRC	ЭM				
	tudents;				
St	cadento,				
St	uuuunus,				
St			-		
S	FirstName	LastName	FirstNameLen	-	astNameLengt
S1	FirstName Alice	Johnson	5	7	astNameLengt
	FirstName Alice Bob	Johnson Smith	5 3	7	astNameLengt
	FirstName Alice Bob Charlie	Johnson Smith Brown	5 3 7	7 5 5	astNameLengt
	FirstName Alice Bob Charlie	Johnson Smith	5 3	7	astNameLengt
•	FirstName Alice Bob Charlie David	Johnson Smith Brown	5 3 7	7 5 5	astNameLengt
▶ SEL	FirstName Alice Bob Charlie David ECT	Johnson Smith Brown Wilson	5 3 7 5	7 5 5 6	
) SEL	FirstName Alice Bob Charlie David ECT UBSTRING(Fi	Johnson Smith Brown Wilson	5 3 7	7 5 5 6	
) SEL	FirstName Alice Bob Charlie David ECT UBSTRING(Fi	Johnson Smith Brown Wilson	5 3 7 5	7 5 5 6	
SEL SI FRC	FirstName Alice Bob Charlie David ECT UBSTRING(Fi	Johnson Smith Brown Wilson	5 3 7 5	7 5 5 6	
SEL SI	FirstName Alice Bob Charlie David ECT UBSTRING(Fi	Johnson Smith Brown Wilson	5 3 7 5	7 5 5 6	
SEL SI FRC	FirstName Alice Bob Charlie David ECT UBSTRING(Fi	Johnson Smith Brown Wilson	5 3 7 5	7 5 5 6	
SEL SI FRC	FirstName Alice Bob Charlie David ECT UBSTRING(Fi DM tudents; FirstThree Ali Bob	Johnson Smith Brown Wilson	5 3 7 5	7 5 5 6	
SEL SI FRC	FirstName Alice Bob Charlie David ECT UBSTRING(Fi DM tudents; FirstThree Ali	Johnson Smith Brown Wilson	5 3 7 5	7 5 5 6	

Programs 13: WAP in SQL for CONDITIONAL statements

CASE Statement: This allows you to perform conditional logic within your SQL queries. WHEN: Specifies the condition to evaluate. THEN: Defines the result to return if the condition is true. ELSE: Specifies a default result if none of the conditions are met. CREATE DATABASE khushi; USE khushi; CREATE TABLE Students (StudentID INT PRIMARY KEY AUTO_INCREMENT, FirstName VARCHAR(50), LastName VARCHAR(50), Score INT); INSERT INTO Students (FirstName, LastName, Score) VALUES ('Alice', 'Johnson', 85), ('Bob', 'Smith', 92), ('Charlie', 'Brown', 75), ('David', 'Wilson', 88), ('Eve', 'Davis', 65); select *from Students; StudentID FirstName LastName Score Alice Johnson 85 1 ۶ 2 Bob Smith 92 3 Charlie Brown 75 4 David Wilson 88 5 Eve Davis 65 NULL NULL NULL NULL SELECT FirstName, LastName, Score, CASE

	WHEN Scor	e >= 90 THE	N 'A'	
	WHEN Scor	e >= 80 THE	N 'B'	
	WHEN Scor	e >= 70 THE	N 'C'	
	WHEN Scor	e >= 60 THE	N 'D'	
	ELSE 'F'			
E	ND AS Grade	e		
FRO	DM			
S	tudents;			
S	tudents; FirstName	LastName	Score	Grade
S		LastName Johnson	Score 85	Grade B
	FirstName			
	FirstName Alice	Johnson	85	В
	FirstName Alice Bob	Johnson Smith	85 92	B A

Program 14: WAP in SQL to learn TRANSCATION.

A transaction in SQL is a sequence of one or more SQL operations that are executed as a single unit of work. Transactions are crucial for maintaining data integrity and consistency in a database, especially when multiple operations need to be treated as a cohesive operation.

CREATE DATABASE BankDB;

USE BankDB;

CREATE TABLE Accounts (

AccountID INT PRIMARY KEY AUTO_INCREMENT,

AccountHolder VARCHAR(100),

Balance DECIMAL(10, 2)

);

INSERT INTO Accounts (AccountHolder, Balance) VALUES ('Alice', 500.00);

INSERT INTO Accounts (AccountHolder, Balance) VALUES ('Bob', 300.00);

-- Start a transaction

START TRANSACTION;

-- Step 1: Deduct money from Account A (Alice)

UPDATE Accounts SET Balance = Balance - 100 WHERE AccountID = 1;

-- Step 2: Add money to Account B (Bob)
UPDATE Accounts SET Balance = Balance + 100 WHERE AccountID = 2;
-- Check Alice's balance after deduction
SELECT Balance FROM Accounts WHERE AccountID = 1;
-- If everything looks good, then commit the transaction
COMMIT; -- Or ROLLBACK; if you see an issue

Program 15

WAP in SQL to learn the concept of TRIGGER.

A **trigger** in SQL is a special kind of stored procedure that automatically executes (or "fires") in response to specific events on a particular table or view. Triggers are typically used to enforce business rules, maintain data integrity, or automate actions within the database without requiring additional application code.

CREATE DATABASE C;

USE C;

-- Create Employees Table

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY AUTO INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Salary DECIMAL(10, 2)

);

-- Create a Log Table to capture changes

CREATE TABLE EmployeeLog (

LogID INT PRIMARY KEY AUTO_INCREMENT,

EmployeeID INT,

Action VARCHAR(50),

LogTime DATETIME DEFAULT CURRENT_TIMESTAMP,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

DELIMITER //

CREATE TRIGGER AfterEmployeeInsert

AFTER INSERT ON Employees

FOR EACH ROW

BEGIN

INSERT INTO EmployeeLog (EmployeeID, Action)

VALUES (NEW.EmployeeID, 'Inserted');

END //

CREATE TRIGGER AfterEmployeeUpdate

AFTER UPDATE ON Employees

FOR EACH ROW

BEGIN

INSERT INTO EmployeeLog (EmployeeID, Action)

VALUES (NEW.EmployeeID, 'Updated');

END //

-- Insert an employee

INSERT INTO Employees (FirstName, LastName, Salary) VALUES ('John', 'Doe', 50000.00);

-- Update an employee's salary

UPDATE Employees SET Salary = 55000.00 WHERE EmployeeID = 1;

-- Check the logs

SELECT * FROM Employees;

	Employ	eeID	FirstName	Last	Vame	Salary
•	1	J	ohn	Doe		55000.00
	NULL	N	VLL	NULL		NULL
SEI	LECT * FI	ROM Em	ployeeLog yeeID A	;; iction	Log	Time
•	1	1	In	serted	2024	4-10-19 11:16
	2	1	Up	odated	2024	4-10-19 11:16
	NULL			LL	NULL	

Program 16

WAP in SQL using UNION and INTERSECT

UNION

- **Purpose**: Combines the result sets of two or more SELECT statements into a single result set.
- **Duplicates**: By default, UNION removes duplicate rows from the result set. If you want to include duplicates, you can use UNION ALL.
- **Requirements**: Each SELECT statement must have the same number of columns in the result set, and the columns must have compatible data types.

-> INTERSECT

- **Purpose**: Returns only the rows that appear in both result sets from two SELECT statements.
- **Duplicates**: INTERSECT automatically removes duplicates, returning only unique rows that are common to both sets.
- **Requirements**: Similar to UNION, each SELECT statement must have the same number of columns and compatible data types.

-- Create a Database

CREATE DATABASE SampleDB;

USE SampleDB;

-- Create First Table: Employees

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY AUTO INCREMENT, FirstName VARCHAR(50), LastName VARCHAR(50)); -- Create Second Table: Managers CREATE TABLE Managers (ManagerID INT PRIMARY KEY AUTO INCREMENT, FirstName VARCHAR(50), LastName VARCHAR(50)); -- Insert Sample Data into Employees INSERT INTO Employees (FirstName, LastName) VALUES ('Alice', 'Smith'); INSERT INTO Employees (FirstName, LastName) VALUES ('Bob', 'Johnson'); INSERT INTO Employees (FirstName, LastName) VALUES ('Charlie', 'Brown'); -- Insert Sample Data into Managers INSERT INTO Managers (FirstName, LastName) VALUES ('Alice', 'Smith'); INSERT INTO Managers (FirstName, LastName) VALUES ('David', 'Wilson'); -- UNION Example SELECT FirstName, LastName FROM Employees UNION SELECT FirstName, LastName FROM Managers; FirstName LastName Alice Smith ٠ Bob Johnson Charlie Brown David Wilson -- INTERSECT Example

SELECT FirstName, LastName FROM Employees INTERSECT SELECT FirstName, LastName FROM Managers; FirstName LastName Alice Smith

Program 17. Write Program for Join, Union & intersection etc.

```
CREATE DATABASE IF NOT EXISTS Sliet_Cse;
USE Sliet_Cse;
CREATE TABLE Employee_1(
Department Name VARCHAR(100) NOT NULL,
Department Id INT NOT NULL,
Employee_Name VARCHAR(50) NOT NULL,
Employee Id INT PRIMARY KEY
);
/*
Department Id for Various Departments:-
1. Computer Science Department :- 101
2. Mechanical Department
                          :- 102
3. Electrical Department
                        :- 103
4. Civil Department
                       :- 104
5. Electronics Department :- 105
*/
INSERT INTO Employee 1
```

 $(Employee_Name, Department_Name, Employee_Id, Department_Id)$

VALUES

("Ansh Girdher", "Computer Science Department", 2211086, 101),

("Lavish Garg", "Computer Science Department", 2211094, 101),

("keshav Khatak","Civil Department",2211097,104),

("Dinesh Kumar","Electronics Department",2211093,105),

("Anil Taak", "Electronics Department", 2211033, 105),

("Manav Garg", "Mechanical Department", 2211096, 102),

("Aditya Chauduary", "Mechanical Department", 2216551, 102),

("Anmoldeep Singh", "Electrical Department", 2214126, 103),

("Dipanshu","Electrical Department",2216553,103);

SELECT * FROM Employee_1 ORDER BY Department_Id ASC;

CREATE TABLE Seminar(

Department_Id INT NOT NULL,

Domain VARCHAR(100) NOT NULL

);

INSERT INTO Seminar

(Department_Id,Domain)

VALUES

(101,"Artificial Intelligence"),

(102,"AutoCAD"),

(103,"MATLAB"),

(104,"Google Sketchup"),

(105,"PIC Microcontroller");

SELECT * FROM Seminar ORDER BY Department_id ASC;

-- Inner Join

SELECT *

FROM Employee_1
INNER JOIN Seminar
ON Employee_1.Department_id = Seminar.Department_id;
Left Join
SELECT *
FROM Employee_1
LEFT JOIN Seminar
ON Employee_1.Department_id = Seminar.Department_id;
Right Join
SELECT *
FROM Employee_1
RIGHT JOIN Seminar
ON Employee_1.Department_id = Seminar.Department_id;
Full Join
SELECT * FROM Employee_1 as a
LEFT JOIN Seminar as b
ON a.Department_id = b.Department_id
UNION
SELECT * FROM Employee_1 as a
RIGHT JOIN Seminar as b
ON a.Department_id = b.Department_id

	base <u>S</u> erver <u>T</u> ools <u>S</u> cripting	<u>H</u> elp				G	9
avigator	Query 1 × SQL File 4* SI	QL File 4*					
HEMAS (%		1 🔂 💿 💿 🐻 Limi	to 1000 rows 👻 📩	V Q 1	IT III III III III III III III III III		
Filter objects	67				, _		
anshi	68 Full Join						
onlinebookstore	69 • SELECT * FROM E	mplovee 1 as a					
sakila sliet cse	70 LEFT JOIN Semina						
Tables	71 ON a.Department	id = b.Department id					
Views	72 UNION						
Stored Procedures Functions	73 SELECT * FROM E	mplovee 1 as a					
	74 RIGHT JOIN Semi						
inistration Schemas	75 ON a.Department	id = b.Department id					
rmation	76						
o object selected	Result Grid 🔢 🚷 Filter Rown	s: Export	Wrap Cell Conten	: <u>IA</u>			
o object selected	Result Grid Image: Comparison of	Department_Id Employe			d Domain		
o object selected	Department_Name Electronics Department	Department_Id Employee 105 Anil Taak	Name Employee_Id 2211033	Department_Id	PIC Microcontroller		
o object selected	Department_Name Electronics Department Computer Science Department	Department_Id Employe 105 Anil Taak 101 Ansh Gird	e_Name Employee_Id 2211033 her 2211086	Department_Id 105 101	PIC Microcontroller Artificial Intelligence		
o object selected	Department_Name Electronics Department Computer Science Department Electronics Department	Department_Id Employee 105 Anil Taak 101 Ansh Gird 105 Dinesh Ku	Employee_Id 2211033 her 2211086 mar 2211093	Department_Id 105 101 105	PIC Microcontroller Artificial Intelligence PIC Microcontroller		
o object selected	Department_Name Electronics Department Computer Science Department	Department_Id Employee 105 Anil Taak 101 Ansh Gird 105 Dinesh Ku	Employee_Id 2211033 her 2211086 mar 2211093 rg 2211094	Department_Id 105 101	PIC Microcontroller Artificial Intelligence		
o object selected	Department_Name Electronics Department Computer Science Department Electronics Department Computer Science Department	Department_Id Employee 105 Anil Taak 101 Ansh Gird 105 Dinesh Ku 101 Lavish Ga	Employee_Id 2211033 her 2211086 mar 2211093 rg 2211094	Department_Id 105 101 105 101	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence		
o object selected	Department_Name Electronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department	Department_Id Employee 105 Anil Taak 101 Ansh Gird 105 Dinesh Ku 101 Lavish Ga	Employee_Id 2211033 her 2211086 mar 2211093 rg 2211094	Department_Id 105 101 105 101	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence		
o object selected	Department_Name Electronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department Result 7 ×	Department_Id Employee 105 Anil Taak 101 Ansh Gird 105 Dinesh Ku 101 Lavish Ga	Employee_Id 2211033 her 2211086 mar 2211093 rg 2211094	Department_Id 105 101 105 101	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence		G
o object selected	Department_Vame Electronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department Result 7 × Output	Department_Id Employee 105 Anil Taak 101 Ansh Gird 105 Dinesh Ku 101 Lavish Ga	Employee_Id 2211033 her 2211086 mar 2211093 rg 2211094	Department_Id 105 101 105 101 102	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence	Duration	Read
o opject selected	Department_Vame Electronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department Mechanical Department Mechanical Department Computer Output Time Action	Department_Id Employee 105 Anil Taak 101 Ansh Gird 105 Dinesh Ku 101 Lavish Ga	E_Mame Employee_Jd 2211033 here 2211086 here 2211096 rg 2211094 rg 2211096	Department_Jd 105 101 105 101 102 Me	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence AutoCAD		Read / Fetch
o opject selected	Department_Vame Electronics Department Computer Science Department Computer Science Department Mechanical Department Mechanical Department Mechanical Department Mechanical Department Action Output Time Action 23 16:39:46 SELECT* FF	Department_Id Employe 105 Anil Taak 101 Ansh Gird 105 Dinesh K. 101 Lavish Ga 102 Manav Ga	E_Mame Employee_Id 2211033 ne 2211086 mar 2211096 rg 2211094 rg 2211096 Seminar ON Employee_1	Department_Id 105 101 105 101 102	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence AutoCAD sssage sw(s) returned	0.000 sec	• Read
o opject selected	Department_Name Bectronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department Result 7 x Output * Time Action Output * Time Action Science Closel CC1* FR 24 16:40:23 SELEC1* FR	Department_Id Employe 105 Ani Task 101 Anh Grid 105 Direch K. 101 Lavish Ga 102 Manav Gr NOM Employee_1 INNER JOIN	211036 2211035 er 2211086 mar 2211096 g 2211094 g 2211094 g 2211096 Seminar ON Employee_1.Dep	Department_Id 105 101 105 101 102	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence AutoCAD sssage wwb) returned ow(s) returned	0.000 sec	Read Fetch C 0.000 s C 0.000 s C 0.000 s
o object selected	Department_Vame Electronica Department Computer Science Department Electronica Department Computer Science Department Mechanical Department Mechanical Department Output Image: Action Output Image: Action Output 23 16:39:46 25 16:41:25 SELECT * FR	Department_Id Employee 105 Anl Taski 101 Anh Grid 105 Direch K. 101 Lavich Ge 102 Manav Ge NOM Employee_1 INNER JOIN SOM Employee_1 LEFT JOIN Se	LName Employee_Ic 221103 Employee_Ic mar 2211093 rg 2211094 rg 2211096 Seminar ON Employee_1.De mnar ON Employee_1.De	Department_Id 105 101 105 101 102 Mee Department	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence AutoCAD sessage ww(s) returned ww(s) returned ww(s) returned	0.000 sec 0.000 sec 0.000 sec	Read / Fetch c / 0.000 sv c / 0.000 sv
-	Department_Vame Electronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department Result 7 × Output Image: Action Output 2 16:39:46 SELECT * FF 2 24 16:40:23 SELECT * FF 2 26 16:42:59 SELECT * FF	Department_Id Employee 105 Ani Taski 101 Anih Grid 105 Dinesh Ku 101 Laveh Ga 102 Manav Gr NOM Employee_1 INNER JOIN SOM Employee_1 LEFT JOIN Se OM Employee_1 RIGHT JOIN S SOM Employee_1 RIGHT JOIN S	1.Name Employee_Jc 221033 mar 221086 mar 221096 rg 221094 rg 221094 rg 221096	Department_Id 105 101 105 101 102 Departmen	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence AutoCAD sessage w(b) returned w(b) returned w(b) returned	0.000 sec 0.000 sec 0.000 sec 0.005 sec	Read Read / Fetch c / 0.000 s c / 0.000 s
No object selected	Department_Vame Electronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department Result 7 × Output Image: Action Output 2 16:39:46 SELECT * FF 2 24 16:40:23 SELECT * FF 2 26 16:42:59 SELECT * FF	Department_Id Employe 105 Ani Taski 101 Anih Grid 105 Diresh K. 101 Lavih Ga 102 Manav Gi NOM Employee_1 INNER JOIN OM Employee_1 LEFT JOIN Se 00M Employee_1 RiGHT JOIN S OM Employee_1 as a LEFT JOIN	1.Name Employee_Jc 221033 mar 221086 mar 221096 rg 221094 rg 221094 rg 221096	Department_Id 105 101 105 101 102 Departmen	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence AutoCAD sessage w(b) returned w(b) returned w(b) returned	0.000 sec 0.000 sec 0.000 sec 0.005 sec	Re Fetch C000 C000 C000 C000 C000 C000

SLIET DBMS × Local	i instance MySQL80 $ imes$	- 0 ×
ile <u>E</u> dit <u>V</u> iew <u>Q</u> uery <u>D</u> ata	abase <u>S</u> erver <u>T</u> ools <u>S</u> cripting <u>H</u> elp	
5 5 6 8 . 5		0
Navigator	Query 1 🗙 SQL File 4* SQL File 4*	
CHEMAS ()	🔌 🔚 🚽 🖉 👰 🕐 196 100 🖏 🐻 Limit to 1000 rows 🔹 🙀 ⊄ 🔍 🕦 💷	
Filter objects	1 • CREATE DATABASE IF NOT EXISTS Sliet Cse;	
anshi anshi	2 • USE Sliet_Cse;	
onlinebookstore	3	
sliet cse	4 •	
▶ 📅 Tables	5 Department_Name VARCHAR(100) NOT NULL,	
Views Stored Procedures	6 Department_Id INT NOT NULL,	
Functions	7 Employee_Name VARCHAR(50) NOT NULL,	
dministration Schemas	8 Employee_Id INT PRIMARY KEY	
nformation	9 ();	
	10	
No object selected	Result Grid 🔢 🚯 Filter Rows: Export: 🎲 Wrap Cal Content: 13	
	Department_Name Department_Id Employee_Name Employee_Id Department_Id Domain	
	Electronics Department 105 Anil Taak 2211033 105 PIC Microcontroller	Result Grid
	▶ Electronics Department 105 Anil Taak 2211033 105 PIC Microcontroller Computer Science Department 101 Ansh Girdher 2211086 101 Artificial Intelligence	
	Electronics Department 105 Ani Taak 2211033 105 PC Morcontroller Computer Science Department 101 Anich Gridher 2211086 101 Anthoid Intelligence Electronics Department 105 Dinesh Kumar 2211093 105 PIC Microcontroller	
	▶ Electronics Department 105 Anil Taak 2211033 105 PIC Microcontroller Computer Science Department 101 Ansh Girdher 2211086 101 Artificial Intelligence	
	Electronics Department 105 Anil Taak 2211033 105 PIC Microcontroller Computer Science Department 101 Andh Girdher 2211086 101 Arthfold Intelligence Electronics Department 105 Dinesh Kumar 2211093 105 PIC Microcontroller Computer Science Department 101 Lavish Garg 2211094 101 Artificial Intelligence	Grid
	▶ Electronics Department 105 Anl Taak 2211033 105 PIC Microantroller Computer Science Department 101 Andh Grinber 2211096 101 Artificial Intelligence Electronics Department 105 Dinesh Kumar 2211093 105 PIC Microcontroller Computer Science Department 101 Lavish Garg 2211094 101 Artificial Intelligence Mechanical Department 101 Lavish Garg 2211094 101 Artificial Intelligence	Result Grid
	▶ Electronics Department 105 Anl Taak 2211033 105 PIC Microantroller Computer Science Department 101 Ansh Girdher 2211086 101 Antificial Intelligence Electronics Department 105 Dinesh Kumar 2211093 105 PIC Microcontroller Computer Science Department 101 Lavish Garg 2211094 101 Artificial Intelligence Mechanical Department 102 Manav Garg 2211096 102 AutoCAD Result 3 ×	Result Grid
	Electronics Department 105 Anith Girdher 2211033 105 PIC Microantholler Computer Science Department 101 Anith Girdher 2211086 101 Antihoil Intelligence Electronics Department 105 Dinesh Kumar 2211093 105 PIC Microcontroller Computer Science Department 101 Lavish Garg 2211094 101 Artificial Intelligence Mechanical Department 102 Manav Garg 2211096 102 AutoCAD Output	Grid
		Read On
	Electronics Department 105 Anit Task 2211033 105 PIC Microantroller Computer Science Department 101 Anah Gräher 2211093 101 Artificial Intelligence Electronics Department 105 Dinesh Kumar 2211093 105 PIC Microantroller Computer Science Department 101 Lavish Garg 2211094 101 Artificial Intelligence Mechanical Department 101 Lavish Garg 2211094 101 Artificial Intelligence Mechanical Department 101 Lavish Garg 2211094 102 AutoCAD Result 3 ×	Read Or Duration / Fetch
	Electronics Department 105 Anit Taak 2211033 105 PIC Microantholer Computer Science Department 101 Ansh Girdher 2211036 101 Antificial Intelligence Electronics Department 105 Dinesh Kumar 2211093 105 PIC Microantholer Computer Science Department 101 Lavich Garg 2211094 101 Antificial Intelligence Mechanical Department 102 Manav Garg 2211096 102 AutoCAD Result 3 × Output Image Image Action Image Image Action	Read Or Duration / Fech 0.000 sec
		Duration / Fetch 0.000 sec
hind life. Session	Electronics Department 105 Anl Tak 2211033 105 PIC Microcontroller Computer Source Department 101 Andh Grither 2211034 101 Artificial Intelligence Electronics Department 105 Dinach Kumar 2211093 105 PIC Microcontroller Computer Soence Department 101 Lavish Garg 2211094 101 Artificial Intelligence Mechanical Department 101 Lavish Garg 2211095 102 Artificial Intelligence Mechanical Department 102 Manav Garg 2211096 102 AutoCAD Restard × Output * Time Action Message © 19 18:37/16 SELECT * FROM Employee_1 INNER JOIN Seminar ON Employee_1.Departmen Error Code: 1146. Table 'employees.mployee_1' doesn't exist © 20 16:37/16 SELECT * FROM Employee_1 INNER JOIN Seminar ON Employee_1.Departmen 9 row(s) infected © 21 16:37/22 SELECT * FROM Employee_1 INNER JOIN Seminar ON Employee_1.Departmen 9 row(s) inturned	Duration / Fetch 0.000 sec 0.000 sec 0.047 sec / 0.000 sec
ibject Info Session	Electronics Department 105 Anit Task 2211033 105 PIC Microcontroller Computer Science Department 101 Anith Gridher 2211096 101 Artificial Intelligence Bectronics Department 105 Dinesh Kumar 2211093 105 Artificial Intelligence Computer Science Department 101 Lavish Garg 2211094 101 Artificial Intelligence Mechanical Department 101 Lavish Garg 2211094 101 Artificial Intelligence Wethanical Department 101 Lavish Garg 2211094 101 Artificial Intelligence Wethanical Department 102 Manav Garg 2211094 101 Artificial Intelligence Virput	Duration / Feach 0.000 sec 0.000 sec 0.000 sec 0.000 sec 0.000 sec

Edit View Query Datab		<u>H</u> elp				Ø –
vigator	Query 1 × SQL File 4* SO	QL File 4*				
EMAS ♥ "Iter objects anshi onlinebookstore sakila silet_cse Tables Views Stored Procedures Functions inistration Schemas	2 ● USE Sliet_Cse; 3 4 ● ○ CREATE TABLE Em; 5 Department_Name 6 Department_Id IV	IF NOT EXISTS Slie	rt_Cse; NULL,	s 🔮 Q, ¶		
no object selected	9); 10 Result Grid III () Filter Rows	s 6	oport: 🙀 Wrap Cell Conter			
	Department_Name Department_Name Department_Name Department Computer Science Department Electronics Department Computer Science Department Mechanical Department	Department_Id Emp 105 Anil T 101 Ansh 105 Dines 101 Lavis	loyee_Name Employee_Io		d Domain PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence AutoCAD	
	Department_Name Electronics Department Computer Science Department Electronics Department Computer Science Department	Department_Id Emp 105 Anil T 101 Ansh 105 Dines 101 Lavis	loyee_Name Employee_Id 'aak 2211033 Girdher 2211086 sh Kumar 2211093 h Garg 2211094	Department_Id 105 101 105 101	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence	Reg
	Department_Name Electronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department	Department_Id Emp 105 Anil T 101 Ansh 105 Dines 101 Lavis	loyee_Name Employee_Id 'aak 2211033 Girdher 2211086 sh Kumar 2211093 h Garg 2211094	Department_Id 105 101 105 101 102	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence	Reg
	Department_Name Electronics Department Computer Science Department Electronics Department Computer Science Department Mechanical Department Mecha	Department_Id Emp 105 Anil 101 Anish 105 Direct 101 Lavis 102 Manz	loyee_Name Employee_Id 'aak 2211033 Girdher 2211086 sh Kumar 2211093 h Garg 2211094	Department_Id 105 101 105 101 105 101 102	PIC Microcontroller Artificial Intelligence PIC Microcontroller Artificial Intelligence AutoCAD sssage w(g) affected	Read Duration / Feth 0.000 sec
	Department_Name Electonics Department Computer Science Department Computer Science Department Computer Science Department Machinical Department Machinical Department @ Action Output # Time Action Qu 16:37:13 USE Silet_Ore Qu 16:37:39 SELECT*FR Qu 21:63:7:93 SELECT*FR Qu 31:63:39:46 SELECT*FR	Department_Id Emp 105 Anili 101 Anih 105 Direc 101 Lavis 102 Mane 100 Mone 100 Mone 00M Employee_1 10M Employee_1 10M Employee_1 10M Employee_1	loyee_IName Employee_In aak 2211033 Gordher 2211036 dh Kumar 2211096 h Gang 2211094 w Gang 2211094 w Gang 2211094	Department_Id 105 101 105 101 102 Mee Orro Department_id 9 ro Department 9 ro	PLC Microcontroller Artificial Intelligence PLC Microcontroller Artificial Intelligence AutoCAD sssge w(b) affected w(b) affected w(b) returned w(b) returned	Read Duration / Fetch

le <u>E</u> dit <u>V</u> iew <u>Q</u> uery <u>D</u> atab	se <u>S</u> erver <u>T</u> ools <u>S</u> cripting <u>H</u> elp	
		Ø 🔲
avigator	Query 1 × SQL File 4* SQL File 4*	
CHEMAS 🚸	🗀 🖬 🗲 👰 🙆 💁 📀 💿 🐻 Limit to 1000 rows 🔹 📩 🚿	0, 1 7
Filter objects	61	
anshi 📃	62 Right Join	
onlinebookstore sakila	63 • SELECT *	
sliet_cse	64 FROM Employee_1	
Tables	65 RIGHT JOIN Seminar	
Views Stored Procedures	<pre>66 ON Employee_1.Department_id = Seminar.Department_id;</pre>	
Functions	67	
ministration Schemas	68 Full Join	
ormation	69 • SELECT * FROM Employee_1 as a	
	70 LEFT JOIN Seminar as b	
No object selected		
No object selected	Result Grid 🔠 😯 Filter Rows: Export: 🏭 Wrap Cell Content: IA	
No object selected	Department_Name Department_Id Employee_Name Employee_Id Depar	rtment_Id Domain
No object selected		rtment_Id Domain Artificial Intelligence Artificial Intelligence
No object selected	Department_Name Department_Id Employee_Name Employee_Id Department > Computer Science Department 101 Lavish Garg 2211094 101	Artificial Intelligence
No object selected	Department_Name Department_Id Employee_Iame Employee_Id Department Computer Science Department 101 Lavish Garg 2211094 101 Computer Science Department 101 Anth Grindher 2211096 101 Mechanical Department 102 Aditya Chauduary 221555 102 Mechanical Department 102 Manare Garg 2211096 102	Artificial Intelligence Artificial Intelligence AutocAD AutocAD
No object selected	Department_Name Department_Id Employee_Iame Employee_Id Department Computer Science Department 101 Lavidh Garg 2211094 101 Computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Adhya Chauduary 2215551 102 Mechanical Department 102 Maney Garg 2211096 102 Electrical Department 103 Departhu 103 Department 103	Artificial Intelligence Artificial Intelligence AutoCAD AutoCAD MATLAB
No object selected	Department_Name Department_Id Employee_Iame Employee_Id Department Computer Science Department 101 Lavidh Garg 2211096 101 Computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Adays Chauduary 2215551 102 Mechanical Department 102 Manav Garg 2211096 102 Electrical Department 103 Dpandhu 2215553 103 Result 5 ×	Artificial Intelligence Artificial Intelligence AutocAD AutocAD
No object selected	Department_Name Department_Id Employee_Id Department Image: Computer Science Department 101 Lavidh Garg 2211096 101 Computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Aditya Chauduary 2216551 102 Mechanical Department 102 Manav Garg 2211096 102 Electrical Department 103 Dipandhu 2215553 103 Result 5 X Output Output Output	Artificial Intelligence Artificial Intelligence AutoCAD AutoCAD MATLAB
No object selected	Department_Name Department_Id Employee_Name Employee_Id Department IC computer Science Department 101 Lavidi Garg 2211094 101 Computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Addra Chauduary 2215551 102 Mechanical Department 102 Addra Chauduary 2215055 102 Electrical Department 103 Dipandhu 2215553 103 Result 5 × Output	Artificial Intelligence Artificial Intelligence AutoCAD AutoCAD MATLAB
No object selected	Department_Name Department_Id Employee_Iame Employee_Id Department Computer Science Department 101 Lavidh Garg 2211096 101 Computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Adhya Chauduary 2216551 102 Mechanical Department 102 Manav Garg 2211096 102 Electrical Department 103 Dipanshu 2216553 103 Result S × Output * * * * f) Action Output * * * *	Intertion Using and Using
No object selected	Department_Name Department_Id Employee_Ia Department Image: Computer Science Department 101 Lavidh Garg 2211094 101 Image: Computer Science Department 101 Lavidh Garg 2211096 101 Mechanical Department 102 Aditya Chauduary 2215551 102 Mechanical Department 102 Manaro Garg 2211096 102 Beckriscial Department 103 Dipandru 2215553 103 Coluput Image: Action Output <	ment_ig_ Comen Artificial Intelligence AutroCAD AutoCAD MATLAB • Read • Read • Read • Read • Duration / Fach • Duration / Fach • 0.047 sec / 0.000 se
No object selected	Department_Name Department_Id Employee_Iame Department_Id Computer Science Department 101 Lavidi Garg 2211094 101 Computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Addys Chaudkary 2215551 102 Mechanical Department 102 Addys Chaudkary 2215055 102 Electrical Department 103 Dipandhu 2216553 103 Result 5 × Output • • • • © 21 1637.22 SELECT * FROM Employee_1 I INVER JOIN Seminar ON Employee_1Department • © 21 1637.23 SELECT * FROM Employee_1 I LEFT JOIN Seminar ON Employee_1Department •	merci ja Goman Artificia Intelligence Artificia Intelligence Artific
No object selected	Department_Name Department_Id Employee_Iame Employee_Id Department IC computer Science Department 101 Lavidh Garg 2211094 101 IC computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Adhya Chauduary 2215055 102 Bechrical Department 102 Manav Garg 2211096 102 Bechrical Department 103 Dipanshu 2216553 103 Result 5 × Output * * * * 0 Utput * * * Time Action * * 0 21 16:37:22 SELECT * FROM Employee_1 LEPT JOIN Seminar ON Employee_1.Department 0 23 16:39:46 SELECT * FROM Employee_1 NINER JOIN Seminar ON Employee_1.Department 0 0 23 16:39:46 SELECT * FROM Employee_1 NINER JOIN Seminar ON Employee_1.Department 0 23 16:39:46 SELECT * FROM Employee_1 NINER JOIN Seminar ON Employee_1.Department 0	A ritical Intelligence Artifical Intelligence Artifical Intelligence Artifical Intelligence AutoCAD AutoCAD MATLAB Read Matlage Duration / Fatch en 9 row(s) returned 0.007 sec / 0.000 sec al 9 row(s) returned 0.000 sec / 0.000 sec
No object selected	Department_Name Department_Id Employee_Iame Employee_Id Department IC computer Science Department 101 Lavidh Garg 2211094 101 IC computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Adhya Chauduary 2215551 102 Beterical Department 102 Manav Garg 2211096 102 Beterical Department 102 Manav Garg 2211096 103 Result 5 × Output • • • • • 01 Action • • • • • • 02 16:3722 SELECT * FROM Employee_1 I INNER JOIN Seminar ON Employee_1 Department • <t< td=""><td>Metrical Intelligence Artificial Intelligence Artificial Intelligence Artificial Intelligence AutoCAD Mattal MATLAB Read Metage Duration / Fatch million 9 row(s) returned 0.000 sec / 0000 sec 0.000 sec / 0000 sec jd. 9 row(s) returned 0.000 sec / 0000 sec</td></t<>	Metrical Intelligence Artificial Intelligence Artificial Intelligence Artificial Intelligence AutoCAD Mattal MATLAB Read Metage Duration / Fatch million 9 row(s) returned 0.000 sec / 0000 sec 0.000 sec / 0000 sec jd. 9 row(s) returned 0.000 sec / 0000 sec
No object selected	Department_Name Department_Id Employee_Iame Employee_Id Department IC computer Science Department 101 Lavidh Garg 2211094 101 IC computer Science Department 101 Andh Girdher 2211096 101 Mechanical Department 101 Andh Girdher 2211096 101 Mechanical Department 102 Adhya Chauduary 2215055 102 Bechrical Department 102 Manav Garg 2211096 102 Bechrical Department 103 Dipanshu 2216553 103 Result 5 × Output * * * * 0 Utput * * * Time Action * * 0 21 16:37:22 SELECT * FROM Employee_1 LEPT JOIN Seminar ON Employee_1.Department 0 23 16:39:46 SELECT * FROM Employee_1 NINER JOIN Seminar ON Employee_1.Department 0 0 23 16:39:46 SELECT * FROM Employee_1 NINER JOIN Seminar ON Employee_1.Department 0 23 16:39:46 SELECT * FROM Employee_1 NINER JOIN Seminar ON Employee_1.Department 0	Metrical Intelligence Artificial Intelligence Artificial Intelligence Artificial Intelligence AutoCAD Mattal MATLAB Read Metage Duration / Fatch million 9 row(s) returned 0.000 sec / 0000 sec 0.000 sec / 0000 sec jd. 9 row(s) returned 0.000 sec / 0000 sec

Program 18. WAP in PL/SQL for adding two numbers.

DECLARE
num1 NUMBER := 10; First number
num2 NUMBER := 20; Second number
sum NUMBER; Variable to store the sum
BEGIN
Calculate the sum
sum := num1 + num2;
Display the result
DBMS_OUTPUT_PUT_LINE('The sum of ' num1 ' and ' num2 ' is: ' sum);
END;

eries.sql	42w5rqw59 🧪		NEW PLSQ	RUN 🕨	:	5
CLARE nun1 NUMBER := 10; First number nun2 NUMBER: = 20; Second number sum NUMBER; Variable to store the sum sitN Calculate the sum sum := num1 + num2; Display the result DBMS_OUTPUT.PUT_LINE('The sum of ' num1 ' ND;	nd ' num2 ' is: ' sum);	STDIN Input for the progra Output: The sum of 10 and 2				

Program 19. WAP in PL/SQL for reversing the number. For example the number is 12345 and reverse number will be 54321.

DECLARE
original_num NUMBER := 12345; Original number to be reversed
reversed_num NUMBER := 0; Variable to store the reversed number
remainder NUMBER; Variable to store the remainder
BEGIN
Display the original number
DBMS_OUTPUT.PUT_LINE('Original number: ' original_num);
Reverse the number
WHILE original_num > 0 LOOP
remainder := MOD(original_num, 10); Get the last digit
reversed_num := (reversed_num * 10) + remainder; Append the last digit to the reversed number
original_num := TRUNC(original_num / 10); Remove the last digit from the original number
END LOOP;
Display the reversed number
DBMS_OUTPUT.PUT_LINE('Reversed number: ' reversed_num);
END;
1

ies.sql 42w5	5rqw59 🧨		NEW PLSQL	RUN 🕨	: :
LARE original_num_NUMBER := 12345; Original_number to be re reversed_num_NUMBER := 0; Variable to store the re remainder NUMBER; Variable to store the re IN Display the original_number DBMS_OUTPUT_PUT_LINE('Original_number: ' original_num); Reverse the number MHILE original_num > 0 LOOP remainder := MOD(original_num, 10); Ge reversed_num := (reversed_num * 10) + remainder; Ap original_num := TRUNC(original_num / 10); Re END LOOP; Display the reversed number DBMS_OUTPUT_PUT_LINE('Reversed number: ' reversed_num); ;	reversed number emainder ; ; tet the Last digit uppend the Last digit to the lemove the Last digit from t	Output: Original num Reversed num		Cti Fentier	

Program 20. WAP in PL/SQL to find the number is even or odd.

```
DECLARE

num NUMBER := 7; -- Number to check (you can change this value)

BEGIN

-- Check if the number is even or odd

IF MOD(num, 2) = 0 THEN

DBMS_OUTPUT.PUT_LINE(num || ' is an Even number.');

ELSE

DBMS_OUTPUT.PUT_LINE(num || ' is an Odd number.');

END IF;

END;

/
```

queries.sql	42w5rqw59 🧪		NEW	PLSQL 🗸	RUN 🕨	:	:::
DECLARE NUM NUMBER := 7; Number to c BEGIN Check if the number is even o IF MOD(num, 2) = 0 THEN DBMS_OUTPUT.PUT_LINE(num ' ELSE END IF; END IF; END (/	is an Even number.');	STDIN Input for the program Output: 7 is an Odd number.	n (Optio	nal)			

Program 21. WAP in PL/SQL to count numbers from 1 to 20.

```
DECLARE

counter NUMBER := 1; -- Start counter from 1

BEGIN

WHILE counter <= 20 LOOP

DBMS_OUTPUT.PUT_LINE(counter); -- Display the current number

counter := counter + 1; -- Increment the counter by 1

END LOOP;

END;

/
```

Prepared by: Dr. Jagdeep Singh, AP(CSE), SLIET Longowal

queries.sql	42w5rqw59 🧪		NEW	PLSQL 🗸	RUN 🕨	:	1
DECLARE counter NUMBER := 1; Start cour	ter from 1	STDIN					
BEGIN		Input for the progr	am (Option	al)			
<pre>WHILE counter <= 20 LOOP DBMS_OUTPUT.PUT_LINE(counter);</pre>	Display the current number	inpactor the progr	antoption	ion y			_
counter := counter + 1;	Increment the counter by 1	Output:					
END LOOP;		1					
ND;	2						
		3					
		4					
		5					
		6					
		8					
		9					
		10					
		11					
		12 13					
		13					
		15					
		16					
		17					
		18					
		19					