Aim: Applying SQL Joins (INNER & OUTER) on a created data base.

Objective: To understand & implement various SQL JOIN operations - including INNER JOIN, LEFT JOIN, RIGHT JOIN & FULL OUTER JOIN. - in order to retrieve & analyze data from multiple rilated

tables in a relational database system.

Aim:	Applying SQL joins (INNER & OUTER) on a
Objective:	To understand & implement various squ JOIN
	operations - including Thurs 1971 1977 1971
	RIGHT JOTH & FULL DUTCH TOTH LEFT JOIN
1	RIGHT JOIN & FULL DUTER JOIN - in order to
	retriève & analyze data from multiple related table
	in a relational database system.
<b>.</b>	
About the	
SEL Joins:	SQL JOIN is used to retrieve data from
	two or more tables based on a related colum
	between them. It allows us to merge rows
	that have a logical relationship making it easier
	to gather information spreaded auross different
	Hables James Miles
Typea:	
٦,	ii) LEFT JOJN
	III) RIGHT JOIN
	iv > FULL OUTER JOIN.
	TOTAL SOLEN SOLEN.
(I	THNER JOIN:
	Returns records that have matching values in
	both tables.
	Excludes nous without matches.

Example:	tustomer -id	une	tity		order_id	twiener -id	מולכו _	
	4				id	-id	amount	,
	2	Allie	MY		707	1	500	
	3	Charle	SF		102	2	300	
	0	Bop	LA		103	_1	100	
5.41	Care .				104	મુ	3250	
Query:	SFIELT	lustom	ero. Lust	Dmer	-name f	rders.or	اندما	
3	Order o	rder am	ount		,	1003.00	~	
				70 7	0711 0-		lustome	
	custome	$\sim id =$	Ondo	1	mer id	10 map	lustome	.22
	- 24		U · UMCS ·	III.STO	mer_id	A CONTRACTOR OF THE PARTY OF TH		
nutout:	metamer_	order	arger-		- A	Tr.		
adjuce	name	1		1000 J	AT PRINTED			
阿五	Alice	101	500		in the			
tr To	Bob	102	300	No.	8	9		
	Alice	103	700	\$		.89 5		
	Charlie	104	250		E E			
12	A.	- 50	155					
	LEFT S	NEO	LEFT 0	UTER	( NEOL	:	- 10.325	
Ś	Returns	الهر اله				Taki.	& match	1
· ·			re man			14014	₩ match	d
	Non ma							11.2
170.0	NULL.		10008	MOKE	The _	what s	ide retur	0
	customer	metamer	app .	1	auger_	to ago wer-	order-	
	id a	name	d		14	id	Janoust	
Example:	1	Alice	NY		101	1	500	_ " ]
	2	Bob	LA		102	2	300	
	3	Chartie	SF	13	103	1	700	
N.	4	David	TX		104	3	250	1.0

ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING & TECHNOLOGY, NAGPUR - 441 108

		TS. orde						
	HKO	0) 10.51	omers_	LEFT :	MIO	Orders	ON lusto	men
	- Just	omer_id	= Dr	ders. lus	tomer	id;		
Output:	Custon	ūst- a	rder_jd	order_			`	
	Alio		tot	500	+		*1	
	Ali		103	700				
	Bob		102	300				
	Cha	Tlie :	104	250	No. of the last of	N		
	Davi	d W	NULL	NULL	65	\$ T		
		6.00 3.00 3.00			伊			
\\	RIGH	IT JOS	N R	THE T	JTER J	; ( NZO	,	
	Retur	ns all	тоще -	t ment	he mig	int to	ble, &	the
				m the			γ	.,
		matchin	4 701	us -tn			side nt	ım
	NUL	THE COLUMN	J. Company	A STATE OF THE STA				
	The Auto-			Ahr Tang	End I e	1.1	a	
xample:	customer.	-name	tity	_1	order_id	instance	amount	
	1	Alice	MY		T0T	1	500	
;	2	Bob	IA		102	2	300	
	3	Charlie	SF	# 	103	1	700	
	20	177			104	3	250	
	SELECT	lusto	mers. w	stomer	name	Orders	order_id	
very;					4			,
rent:		order	mount	1 -				

	The second second			STATE OF THE PARTY	E STORY OF THE RESIDENCE		-
Culput :	inthony of	septe.	annant				
	Alice	101	500			and the state of t	
	Bub	_tot_	300		· Corre and the second of the		
	Alice	LOE	100				
1	Charlie	701	920			1	
		,					
	lan.			: (NEUE		11	
	Reluma	all r	tords -		laldes	IA III W	ull'e_
	in places	wher	e DIA	a_malch	_dom'nt	entities :	
Example:	tis twinter	tuskense.	apl	Jugie 14	(nepages	erder.	
	<u>3</u> L	Alke	MY	101	E	500	
	9.	Bob	In the	102	9	300	
37	3	Charle	D= W	40%	T	00 F	
	4	David	TX	404	7	250	
		1		dob	2	150	
Overy:	SEE	Who Mrs.	excess when by	and and			
U	SELECT	Custome	es. tust	umer_name	Order	a orderi	4
	Orders. v				R. and the same place		
	FROM Customers toll to JOIN Orders UN Customers.					mem.	
	Curtomer id = Orders, customer 11/4						
	plu +						
				No.			

ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING & TECHNOLOGY, NAGPUR : 441 108

Conclu	: miz	This SQL acros	experion con services of the s	oment U oper outtiple E wo	demonstrate utions to tables.  k is es ment sy	d the query Undersi	use of data	Spread how each lutional
7.					welcut con	vels:		i de de la companya d
- A		2 to 1	1	101		111/10		
	1 0	91		124	N. S.	- Should -		
				- 2 h		454		
	4	4	4	id	- XLL	- H	- 4	
	F.1 - 1	tier	1 de 1	he story	112	provided !	bi	Lalg meat.
	stelde	1 10	elielly -	-U.L.	u with tell			
- 111	(1-1 - c) t	144	es Hali.	illich in	at tower	1/18 20 Nest 20	enter.	
				(HL)	1100 110	LE MAN	1111	
	54, -9*- AND (II							eine of the land
W. M					Topic and a supplement of the	Ham from t	S	
	and the second of the second o	z v symere			A STATE OF THE STA	A. A.	-1-25	The second secon
	مقرأ ورامايد			The second	Ludi L	aL L	and bear	
1					Coper son s		12147	. Majimil.

## 1. SQL JOIN

The JOIN operation in SQL is used to combine rows from two or more tables based on a related column. It helps in fetching data that is stored across multiple tables.

## Types of JOINs in SQL

# 1. INNER JOIN (Default JOIN)

- Returns only the records where there is a match in both tables.
- If there is no match, rows from both tables are excluded.

## **Example:**

Consider the following tables:

## **Customers Table**

customer_id	customer_nam e	city
1 -	Alice	NY
2	Bob	LA
3	Charlie	SF

## **Orders Table**

order_id	customer_i d	order_amount
101	1	500
102	2	300
103	1	700
104	3	250

## Query (Using INNER JOIN):

SELECT eustomers.customer\_name, Orders.order\_id, Orders.order\_amount

## FROM Customers

INNER JOIN Orders ON Customers.customer\_id = Orders.customer\_id;

#### **Output:**

customer_name	order_i d	order_amount
Alice	101	500
Alice	103	700
Bob	102	300
Charlie	104	250

## 2. LEFT JOIN (LEFT OUTER JOIN)

- Returns all rows from the left table and only matching rows from the right table.
- If there is no match, the right table returns NULL.

## **Query (Using LEFT JOIN):**

SELECT Customers.customer\_name, Orders.order\_id, Orders.order\_amount

**FROM Customers** 

LEFT JOIN Orders ON Customers.customer\_id = Orders.customer\_id;

## Output:

customer_name	order_i d	order_amount
Alice	101	500
Alice	103	700
Bob	102	300
Charlie	104	250
David	NULL	NULL

(If there was a customer in Customers who didn't place an order, their order fields would be NULL.)

## 3. RIGHT JOIN (RIGHT OUTER JOIN)

- Returns all rows from the right table and only matching rows from the left table.
- If there is no match, the left table returns NULL.

## Query (Using RIGHT JOIN):

SELECT Customers.customer\_name, Orders.order\_id, Orders.order\_amount FROM Customers

RIGHT JOIN Orders ON Customers.customer\_id = Orders.customer\_id;

#### 4. FULL JOIN (FULL OUTER JOIN)

- Returns all rows when there is a match in either table.
- If there is no match, NULL values are returned for missing matches.

#### Query (Using FULL JOIN):

SELECT Customers.customer\_name, Orders.order\_id, Orders.order\_amount FROM Customers

FULL JOIN Orders ON Customers.customer\_id = Orders.customer\_id;

#### 5. CROSS JOIN

 Returns the Cartesian product (all possible combinations of rows) between two tables.

## Query (Using CROSS JOIN):

SELECT Customers.custorper name, Orders.order\_id, Orders.order\_amount

**FROM Customers** 

CROSS JOHN Orders;

#### 2 SQL UNION

The UNION operation in SQL is used to combine the results of two or more SELECT statements. Unlike JOIN, which combines columns from multiple tables, UNION stacks rows from multiple queries into a single result set.

#### Rules for UNION

- · The number of columns in all queries must be the same.
- The data types of corresponding columns must be compatible.
- UNION automatically removes duplicate rows unless UNION ALL is used.

## 1. UNION (Removes Duplicates)

**Example**:

Employees\_DeptA Table

emp_id	emp_name	department
1	John	IT
2	Alice	IT the state of

**Employees\_DeptB Table** 

emp_id	emp_name	department
3	Bob	HR
4	Charlie	HR
1	John	IT

## Query (Using UNION):

SELECT emp\_id, emp\_name, department FROM Employees\_DeptA UNION

SELECT emp\_id, emp\_name, department FROM Employees\_DeptB;

## **Output:**

emp_id	emp_name	department
1	John	IT
2	Alice	ĺΤ
3	Bob	HR
4	Charlie	HR

Explanation:

The duplicate "John" from Employees\_DeptB is removed.

## 2. UNION ALL (Includes Duplicates)

## **Query (Using UNION ALL)**:

SELECT emp\_id, emp\_name, department FROM Employees\_DeptA

**UNION ALL** 

SELECT emp\_id, emp\_name, department FROM Employees\_DeptB;

**Output:** 

emp_id	emp_name	department
1	John	IT
2	Alice	IT
3	Bob	HR
4	Charlie	HR
1	John	ΙΤ

## **Key Differences Between JOIN and UNION**

Feature	JOIN	UNION	
Purpose	Combines columns from multiple tables based on a condition.	Combines rows from multiple queries into a single result set.	
Number of Columns	Can include different columns from different tables.	All queries must have the same number of columns.	
Matching Condition	Uses ON condition to match rows.	No matching condition is required.	
Duplicate Handling	Keeps all matching rows.	UNION removes duplicates, UNION ALL keeps them.	
Example Use Fetching related data from multiple case tables (e.g., customers & orders).		Merging similar datasets (e.g., employees from different departments).	

Mulaut :	lustomer -name	arder	amount		
Output:	Alice	101	500		
	Bob	102	300		
	Alice	103	700		
- 18	Charlie	104	250	•	
	David	NULL	NULL		
1.1	NULL	105	150		
		Ŧ			
Procedure:					
_	Create two tables (eg lustomera & Orders) with a				
	Common key (lustomor id)				
ζü	Inserts sample records for each table.				
	Exerute the SQL queries for INHER JOJN, LEFT				
	JOIN REGHT JOIN, & FULL OVIER JOIN.				
. 1	Observe & records the results.				
	Analyze how different types of JOIN's affect the				
	result set.				
Conclusion:	This experiment demonstrated the use of various				
ì	SUL JUIN operations to query data spread				
	accross multiple tables. Understanding how each				
	JUIN TYPE works is essential in mational				
	dalchasse muangements system				
	(At) Ruduli 24/4/25				
				24/4/25	
		4.5%			

ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING & TECHNOLOGY, NAGPUR - 441 108