

SE COMP - A		Roll number : 9913	
Experiment no. : 7		Date of Implementation: 26/ 03/ 2024	
Aim : To implement Nested Sub-queries in SQL			
Tool Used : PostgreSQL/ Mysql			
Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database			
Rubrics for assessment of Experiment:			
Indicator	Poor	Average	Good
Timeliness <ul style="list-style-type: none"> Maintains assignment deadline (3) 	Assignment not done (0)	One or More than One week late (1-2)	Maintains deadline (3)
Completeness and neatness <ul style="list-style-type: none"> Complete all parts of assignment(3) 	N/A	< 80% complete (1-2)	100% complete (3)
Originality <ul style="list-style-type: none"> Extent of plagiarism(2) 	Copied it from someone else(0)	At least few questions have been done without copying(1)	Assignment has been solved completely without copying (2)
Knowledge <ul style="list-style-type: none"> In depth knowledge of the assignment(2) 	Unable to answer 2 questions(0)	Unable to answer 1 question (1)	Able to answer 2 questions (2)
Assessment Marks :			
Timeliness			
Completeness and neatness			
Originality			
Knowledge			
Total			
Total : (Out of 10)			

Teacher's Sign :		
	EXPERIMENT 7	Nested subqueries in SQL
	Aim	To implement nested sub-queries in SQL
	Tools	PostgreSQL/Mysql

Procedure	<p>Use the tables created in the previous experiments and Perform the following queries using nested sub-queries.</p> <p>Client_master (client_no, name, address, city, pincode, state, bal_due)</p> <p>Product_master (product_no, description, profit_percentage, unit_measure, qty_on_hand, reorder_level, sell_price, cost_price)</p> <p>Sales_order(order_no, order_date, client_no, dely_Addr, salesman_no, dely_type, billed_yn, dely_date, order_status)</p> <p>Sales_order_details(order_no, product_no, qty_ordered, qty_disp, product_rate)</p> <p>1. Find the product no. and description of non-moving products i.e. products not being sold.</p> <pre>SELECT product_no, description FROM product_master WHERE product_no NOT IN (SELECT DISTINCT product_no FROM sales_order_details);</pre> <table><thead><tr><th></th><th>product_no [PK] character varying (6)</th><th>description character varying (15)</th></tr></thead><tbody><tr><td>1</td><td>P004</td><td>Keypad</td></tr><tr><td>2</td><td>P005</td><td>Printer</td></tr></tbody></table> <p>2. Find the customer name, address for the client who has placed order no 'O191'</p> <pre>SELECT name, address FROM client_master WHERE client_no = (SELECT client_no FROM salesorder WHERE order_no = 'ORD191');</pre> <div><div>Data output Messages Notifications</div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><table><thead><tr><th>name character varying (20)</th><th>address character varying (30)</th></tr></thead><tbody></tbody></table></div></div> <p>3. Find the clients names who have placed orders before the month of May'96</p> <pre>SELECT name FROM client_master WHERE client_no IN (SELECT cLiENT_no FROM salesorder WHERE EXTRACT(MONTH FROM order_date) < 5 AND EXTRACT (YEAR FROM order_date) = 1996);</pre> <div><div><div></div><div></div><div></div><div></div><div></div><div></div></div><table><thead><tr><th>name character varying (20)</th></tr></thead><tbody></tbody></table></div>		product_no [PK] character varying (6)	description character varying (15)	1	P004	Keypad	2	P005	Printer	name character varying (20)	address character varying (30)	name character varying (20)
	product_no [PK] character varying (6)	description character varying (15)											
1	P004	Keypad											
2	P005	Printer											
name character varying (20)	address character varying (30)												
name character varying (20)													

4. Find out if the product '1.44 Drive' has been ordered by any client and print the client_no, name to whom it was sold

```
SELECT client_no, name FROM client_master
WHERE client_no
IN
(SELECT client_no FROM sales_order_details
WHERE product_no = (SELECT product_no
FROM product_master
WHERE description = '1.44 Drive'));
```

	client_no [PK] character varying (6)		name character varying (20)				

5. Find the names of clients who have placed orders worth Rs. 10000 or more

```
SELECT name FROM client_master
WHERE client_no
IN
(SELECT client_no FROM sales_order_details
GROUP BY client_no
HAVING
SUM(qty_ordered * product_rate) >= 10000);
```

	name character varying (20)				

6. Retrieve all the orders placed by a client named 'Rahul Desai' from the sales_order table.

```
SELECT order_no FROM salesorder
WHERE
client_no = (SELECT client_no
FROM client_master
WHERE name = 'Rahul Desai');
```

	order_no character varying (6)				

7. Retrieve name, address, city of all the clients who have placed an order through salesman no 's001'.

```
SELECT name, address, city FROM client_master
WHERE client_no IN (SELECT client_no FROM salesorder WHERE salesman_no = 's001');
```

	name character varying (20)	address character varying (30)	city character varying (15)
1	John	Something1	Ahmednagar

8. Find out all the products that are not being sold from the product_master table, based on the products actually sold as shown in the sales_order_details table.

```
SELECT product_no FROM product_master
WHERE product_no
NOT IN
(SELECT product_no
FROM sales_order_details);
```

	product_no [PK] character varying (6)
1	P004
2	P005

9. Retrieve the product numbers, their description and the total quantity ordered for each product.

```
SELECT product_master.product_no, description,
SUM(qty_ordered)
AS total_qty_ordered
FROM sales_order_details
JOIN product_master
ON sales_order_details.product_no = product_master.product_no
GROUP BY product_master.product_no, description;
```

	product_no [PK] character varying (6)	description character varying (15)	total_qty_ordered numeric
1	P002	Hard disk	5
2	P003	Processor	8
3	P001	Laptop	25

Post Lab Questions:	<ol style="list-style-type: none"> 1. What is incremental Update? In database management systems (DBMS), an incremental update refers to updating data by only modifying the specific changes made since the last update, rather than reprocessing the entire dataset. This method is efficient for large datasets where only a portion of the data has been altered. 2. Explain is use of on delete cascade and on update cascade with suitable example? In DBMS, the "ON DELETE CASCADE" and "ON UPDATE CASCADE" are referential actions that can be specified when defining foreign key constraints in relational databases. Here is how they are used: <ul style="list-style-type: none"> - <u>ON DELETE CASCADE</u>: When a record in the parent table is deleted, all related records in the child table will be automatically deleted. This maintains referential integrity by removing dependent records when the referenced record is removed. It ensures that there are no orphaned rows in the child table(s). - <u>ON UPDATE CASCADE</u>: When a record in the parent table is updated, the matching records in the child table will be automatically updated. This action ensures that changes made to the parent table are reflected in the child table, maintaining consistency between the two tables.
----------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------