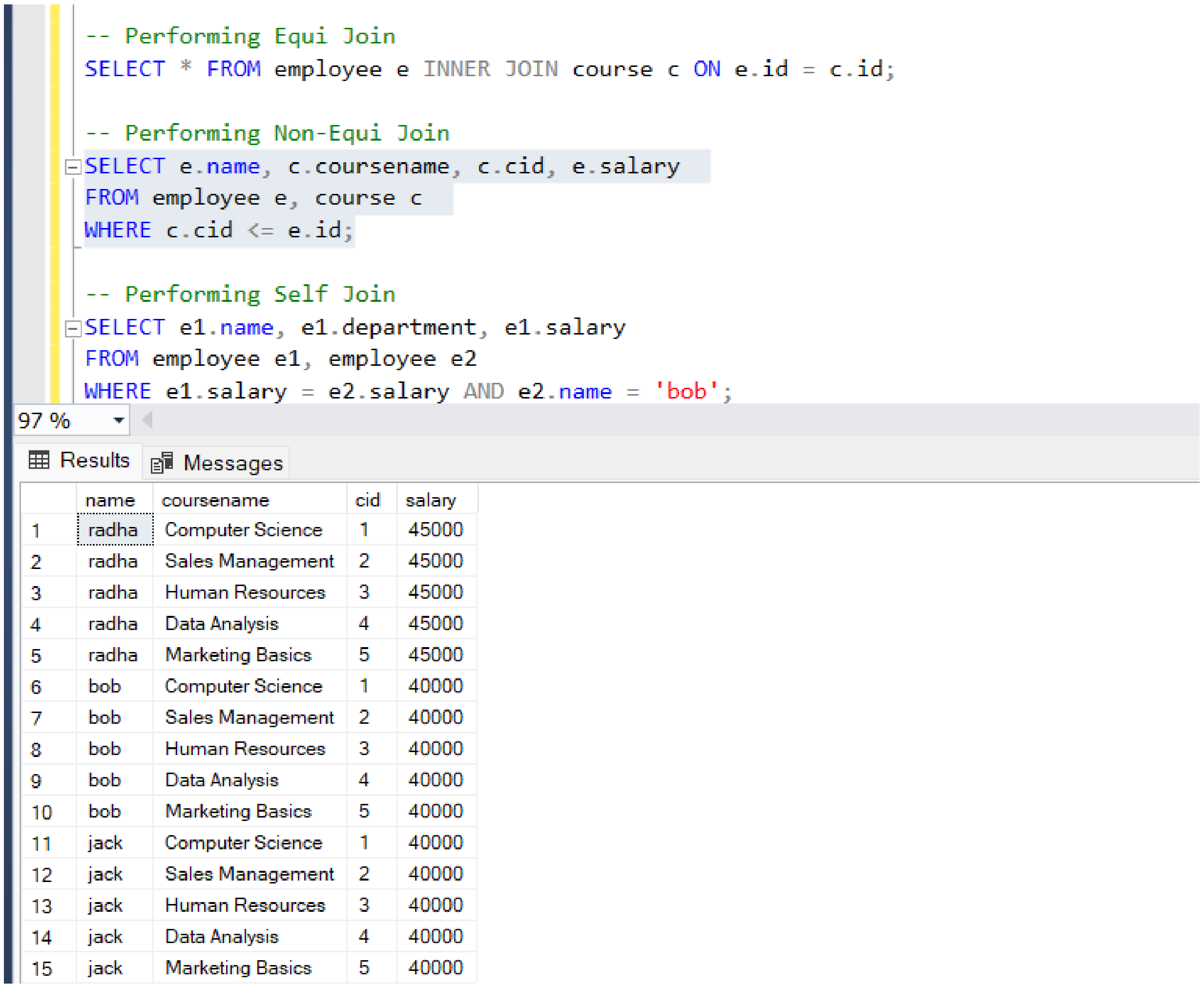
Subqueries, Corelated Subquery & Set Operators

Name: Aathirainathan P

Date: 08-11-2024

---

1. Equi-Join,Non-Equi Join,Self Join:



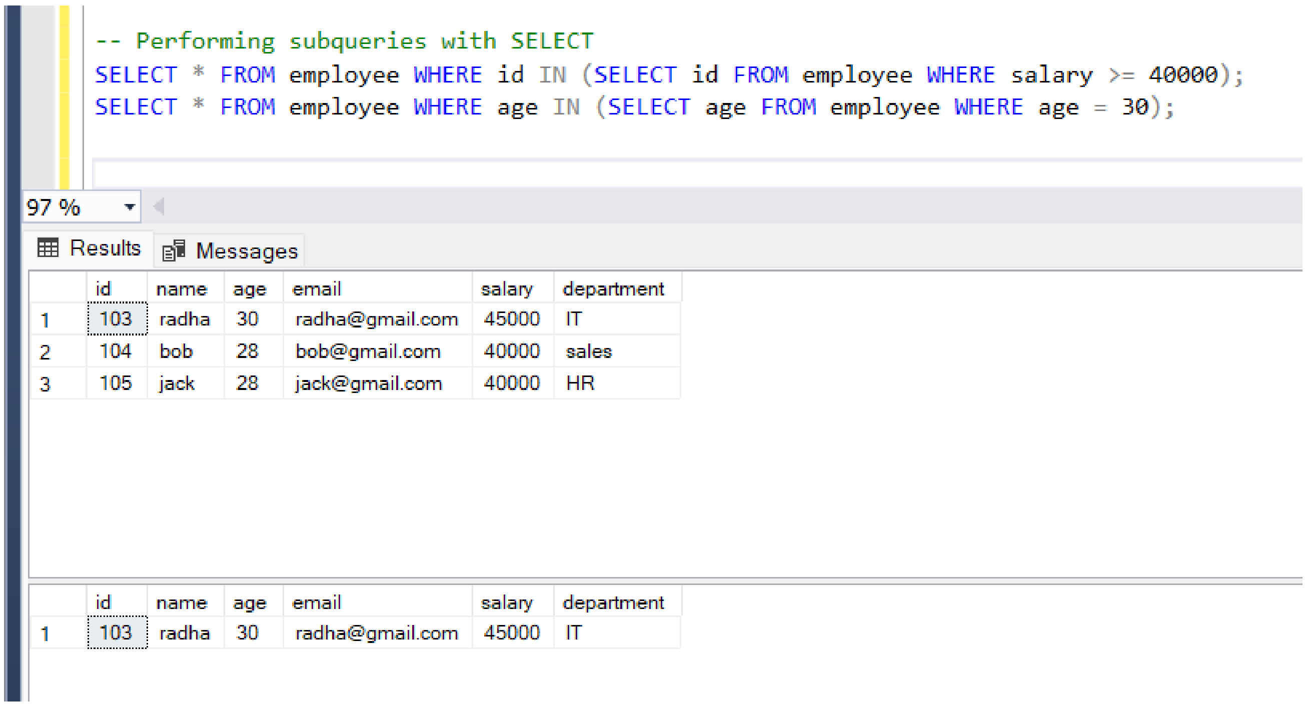
```
-- Performing Equi Join
SELECT * FROM employee e INNER JOIN course c ON e.id = c.id;

-- Performing Non-Equi Join
SELECT e.name, c.coursename, c.cid, e.salary
FROM employee e, course c
WHERE c.cid <= e.id;

-- Performing Self Join
SELECT e1.name, e1.department, e1.salary
FROM employee e1, employee e2
WHERE e1.salary = e2.salary AND e2.name = 'bob';
```

97 %

Results | Messages

	name	coursename	cid	salary
1	radha	Computer Science	1	45000
2	radha	Sales Management	2	45000
3	radha	Human Resources	3	45000
4	radha	Data Analysis	4	45000
5	radha	Marketing Basics	5	45000
6	bob	Computer Science	1	40000
7	bob	Sales Management	2	40000
8	bob	Human Resources	3	40000
9	bob	Data Analysis	4	40000
10	bob	Marketing Basics	5	40000
11	jack	Computer Science	1	40000
12	jack	Sales Management	2	40000
13	jack	Human Resources	3	40000
14	jack	Data Analysis	4	40000
15	jack	Marketing Basics	5	40000

2. Subqueries with select:



```
-- Performing subqueries with SELECT
SELECT * FROM employee WHERE id IN (SELECT id FROM employee WHERE salary >= 40000);
SELECT * FROM employee WHERE age IN (SELECT age FROM employee WHERE age = 30);
```

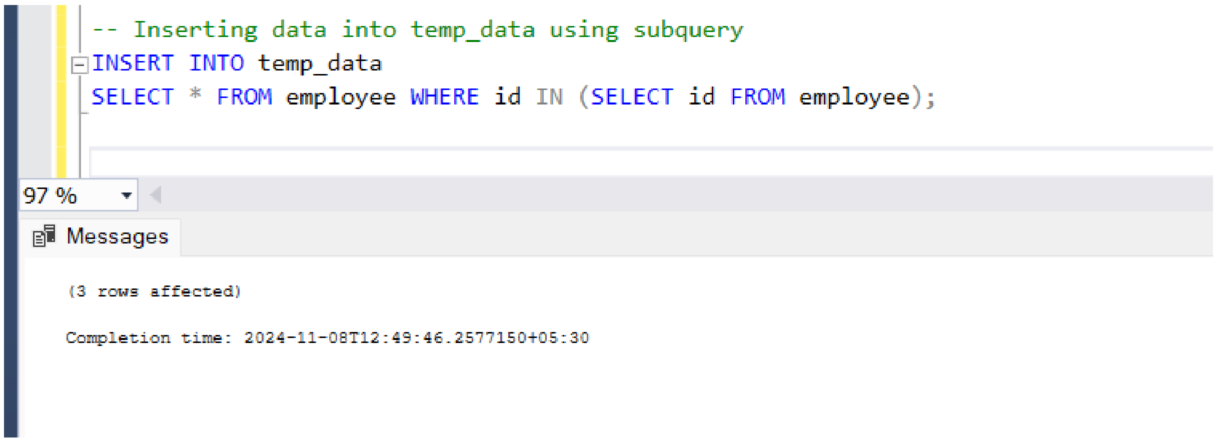
97 %

Results | Messages

	id	name	age	email	salary	department
1	103	radha	30	radha@gmail.com	45000	IT
2	104	bob	28	bob@gmail.com	40000	sales
3	105	jack	28	jack@gmail.com	40000	HR

	id	name	age	email	salary	department
1	103	radha	30	radha@gmail.com	45000	IT

3. Insert using Joins:



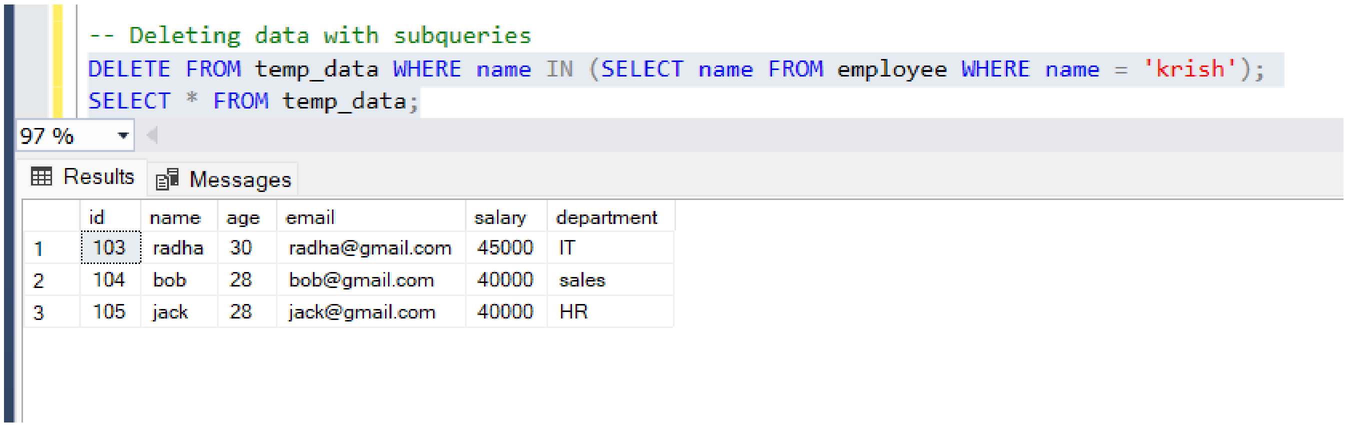
```
-- Inserting data into temp_data using subquery
INSERT INTO temp_data
SELECT * FROM employee WHERE id IN (SELECT id FROM employee);
```

97 %

Messages

```
(3 rows affected)

Completion time: 2024-11-08T12:49:46.2577150+05:30
```

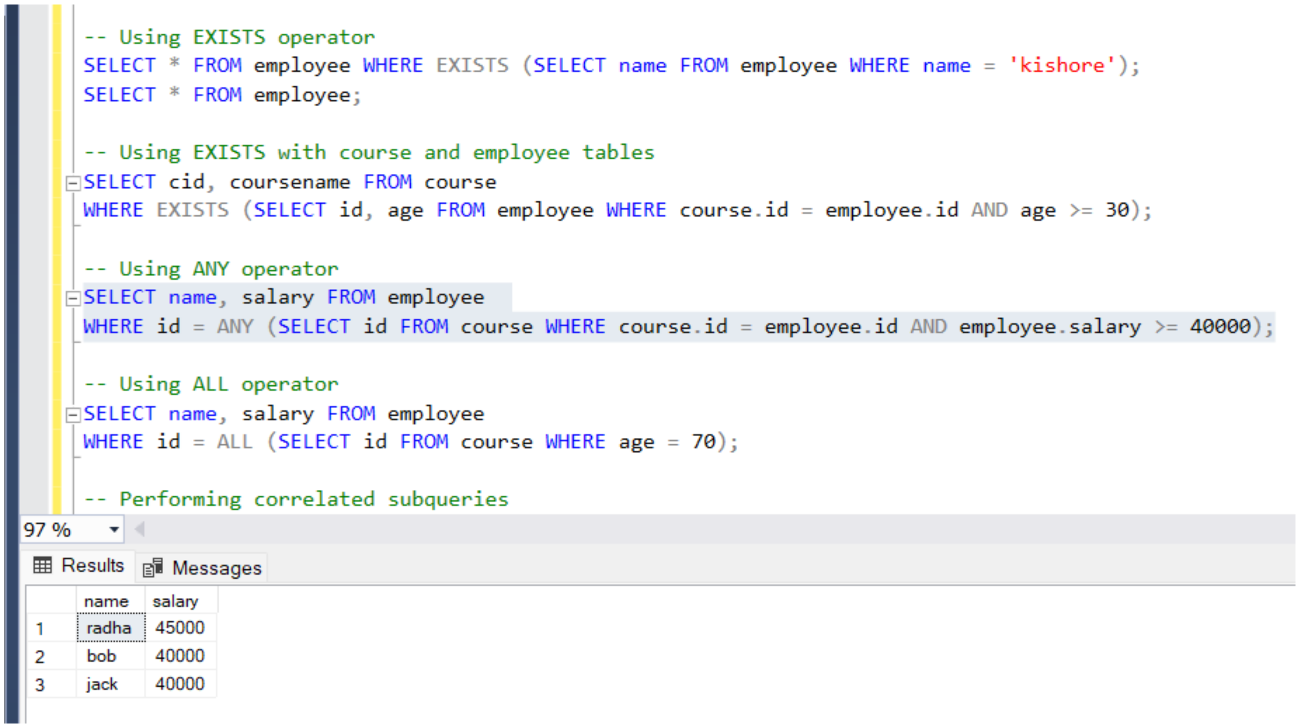
4. Delete Using Subquery:



```
-- Deleting data with subqueries
DELETE FROM temp_data WHERE name IN (SELECT name FROM employee WHERE name = 'krish');
SELECT * FROM temp_data;
```

	id	name	age	email	salary	department
1	103	radha	30	radha@gmail.com	45000	IT
2	104	bob	28	bob@gmail.com	40000	sales
3	105	jack	28	jack@gmail.com	40000	HR

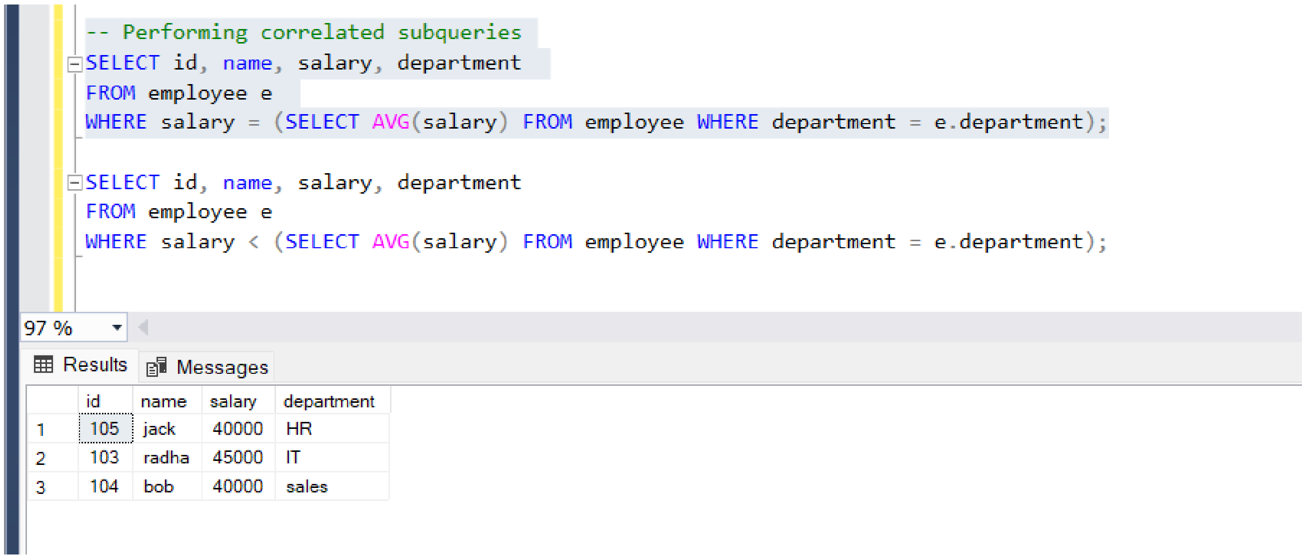
5. Exists,Any,All:



```
-- Using EXISTS operator
SELECT * FROM employee WHERE EXISTS (SELECT name FROM employee WHERE name = 'kishore');
SELECT * FROM employee;

-- Using EXISTS with course and employee tables
SELECT cid, coursename FROM course
WHERE EXISTS (SELECT id, age FROM employee WHERE course.id = employee.id AND age >= 30);

-- Using ANY operator
SELECT name, salary FROM employee
WHERE id = ANY (SELECT id FROM course WHERE course.id = employee.id AND employee.salary >= 40000);

-- Using ALL operator
SELECT name, salary FROM employee
WHERE id = ALL (SELECT id FROM course WHERE age = 70);

-- Performing correlated subqueries
```

	name	salary
1	radha	45000
2	bob	40000
3	jack	40000

6. Correlated Subqueries:



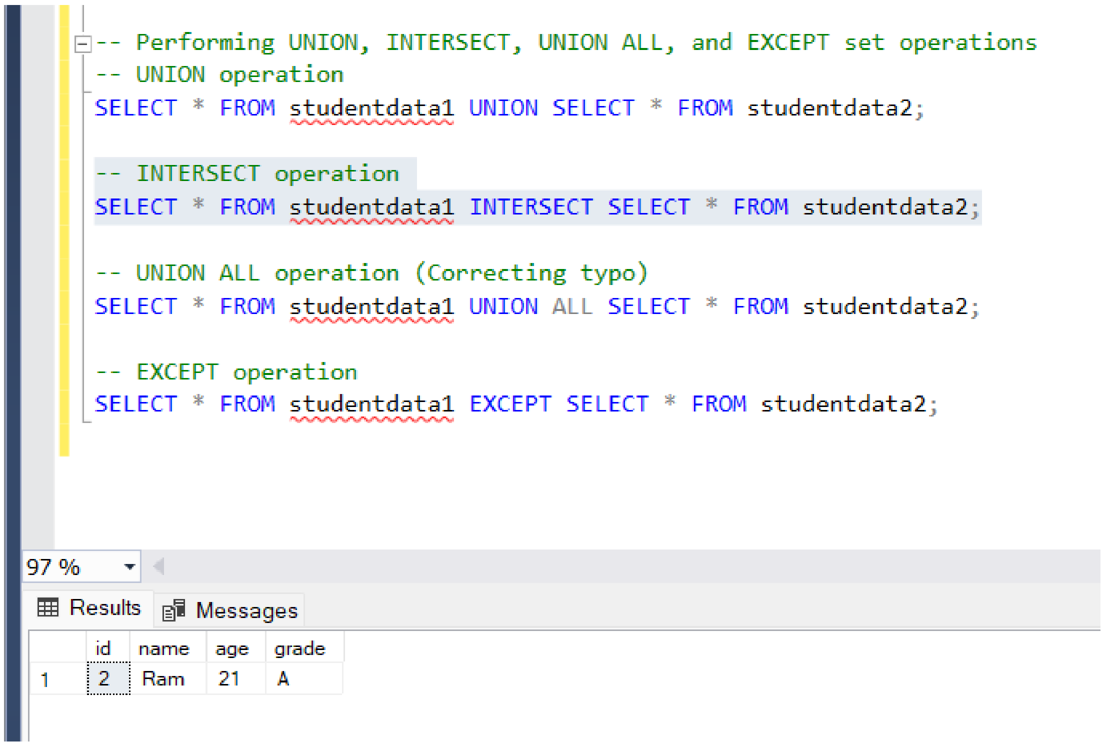
```
-- Performing correlated subqueries
SELECT id, name, salary, department
FROM employee e
WHERE salary = (SELECT AVG(salary) FROM employee WHERE department = e.department);

SELECT id, name, salary, department
FROM employee e
WHERE salary < (SELECT AVG(salary) FROM employee WHERE department = e.department);
```

97 %

Results | Messages

	id	name	salary	department
1	105	jack	40000	HR
2	103	radha	45000	IT
3	104	bob	40000	sales

7. UNION, INTERSECT, UNION ALL, and EXCEPT



```
-- Performing UNION, INTERSECT, UNION ALL, and EXCEPT set operations
-- UNION operation
SELECT * FROM studentdata1 UNION SELECT * FROM studentdata2;

-- INTERSECT operation
SELECT * FROM studentdata1 INTERSECT SELECT * FROM studentdata2;

-- UNION ALL operation (Correcting typo)
SELECT * FROM studentdata1 UNION ALL SELECT * FROM studentdata2;

-- EXCEPT operation
SELECT * FROM studentdata1 EXCEPT SELECT * FROM studentdata2;
```

97 %

Results | Messages

	id	name	age	grade
1	2	Ram	21	A