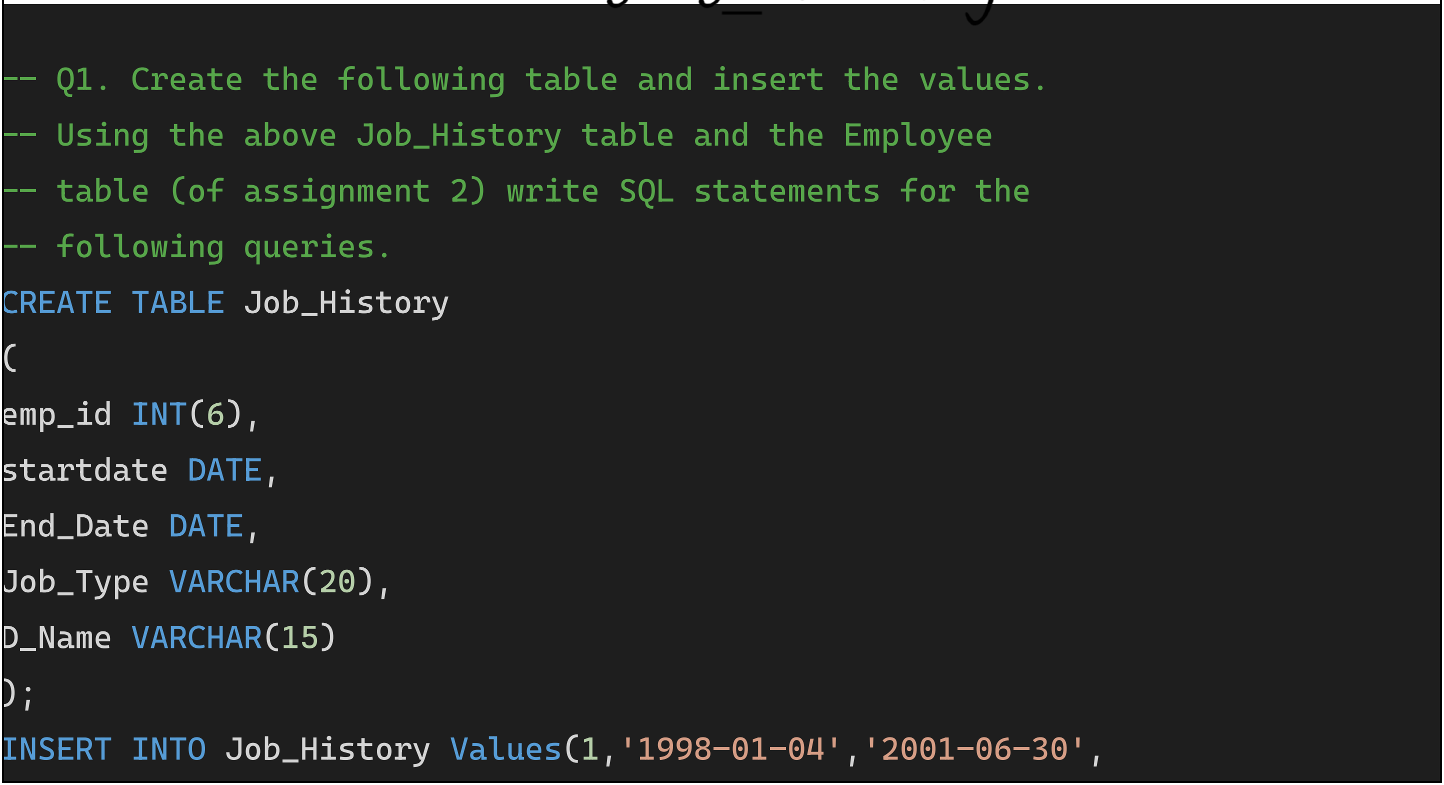
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

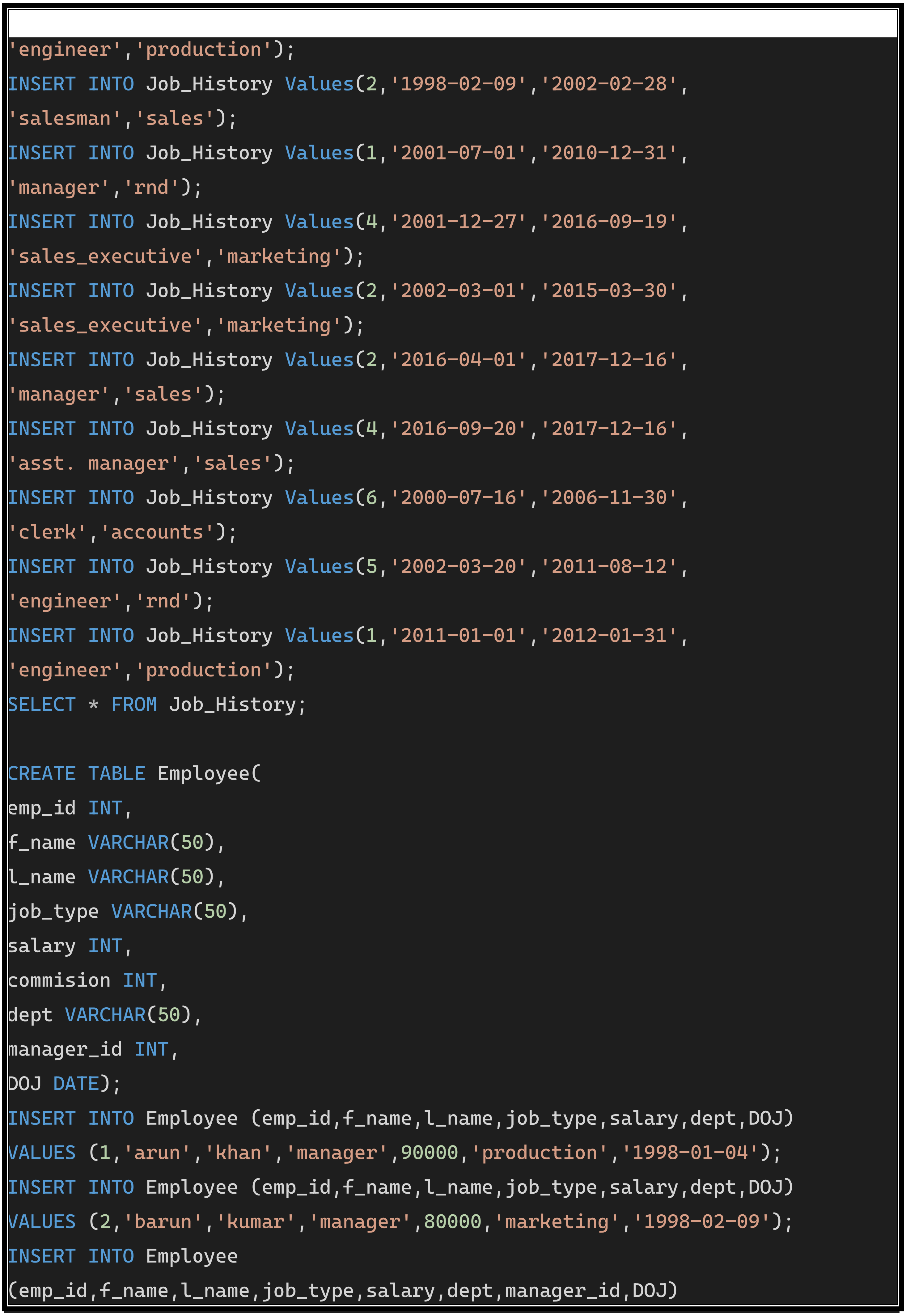
DBMS LAB ASSIG 7

- Name :HITU RAJ
- Roll no. :2005025
- Branch :CSE

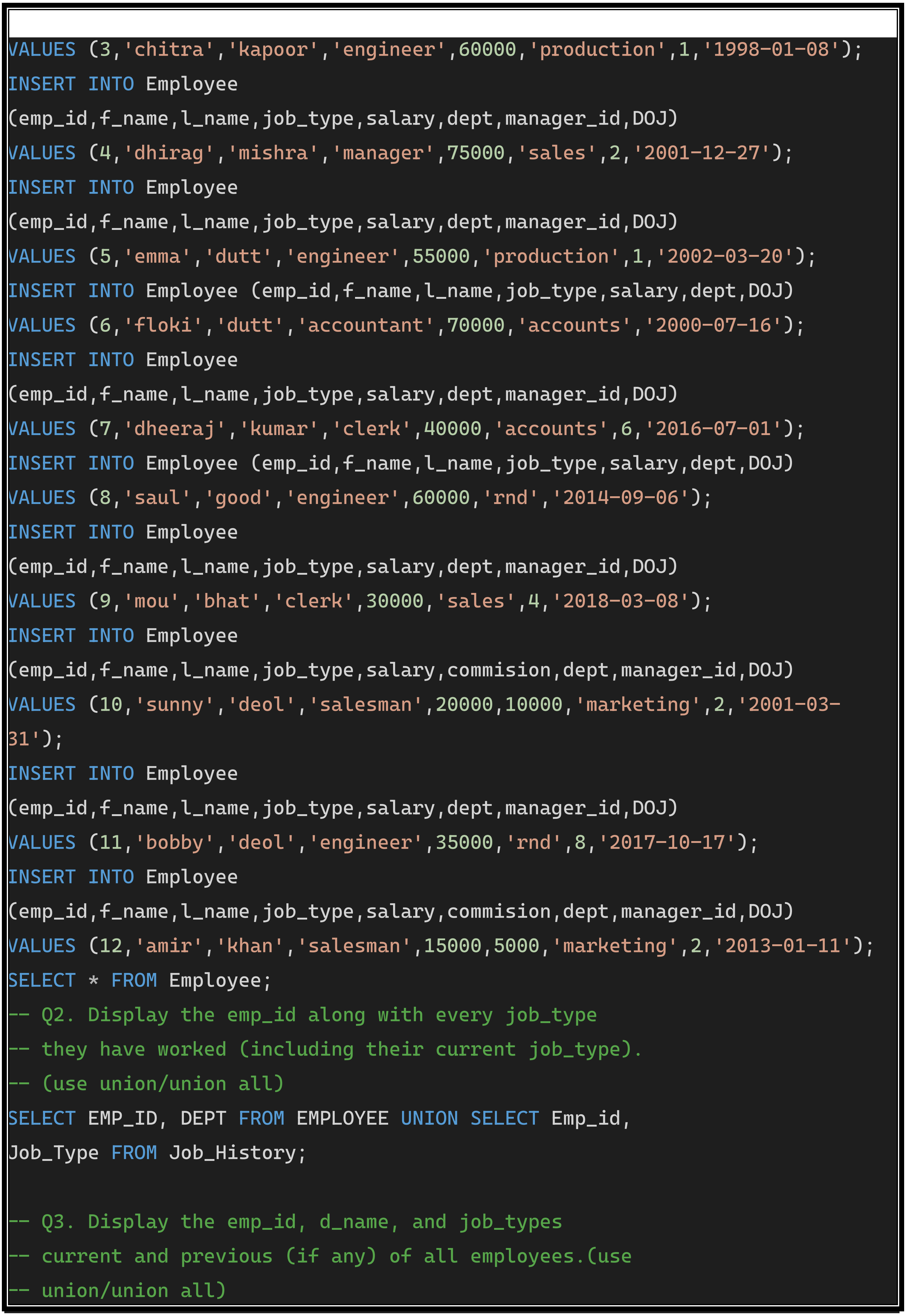
2005025_Hitu raj



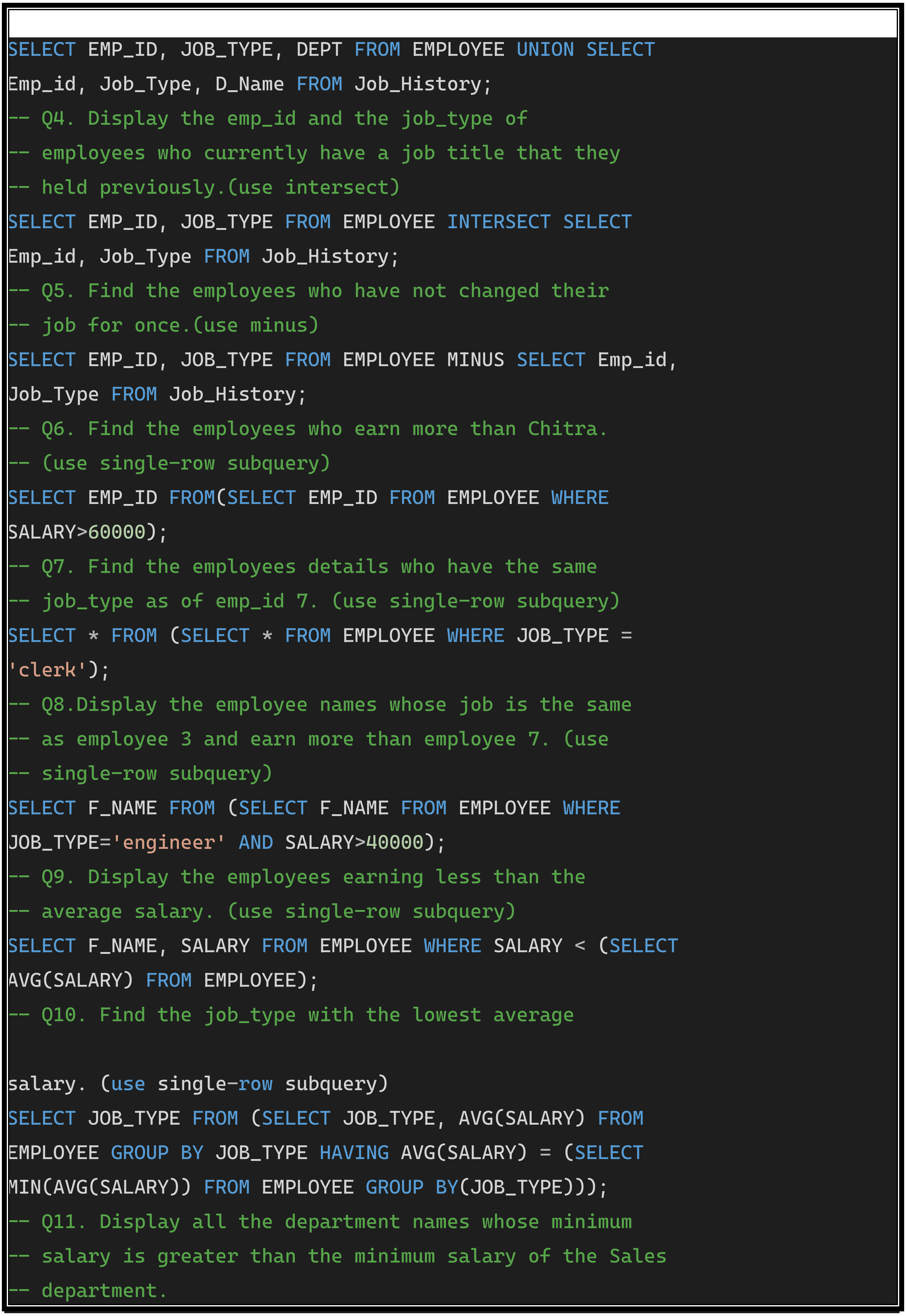
```
-- Q1. Create the following table and insert the values.
-- Using the above Job_History table and the Employee
-- table (of assignment 2) write SQL statements for the
-- following queries.
CREATE TABLE Job_History
(
emp_id INT(6),
startdate DATE,
End_Date DATE,
Job_Type VARCHAR(20),
D_Name VARCHAR(15)
);
INSERT INTO Job_History Values(1,'1998-01-04','2001-06-30',
```



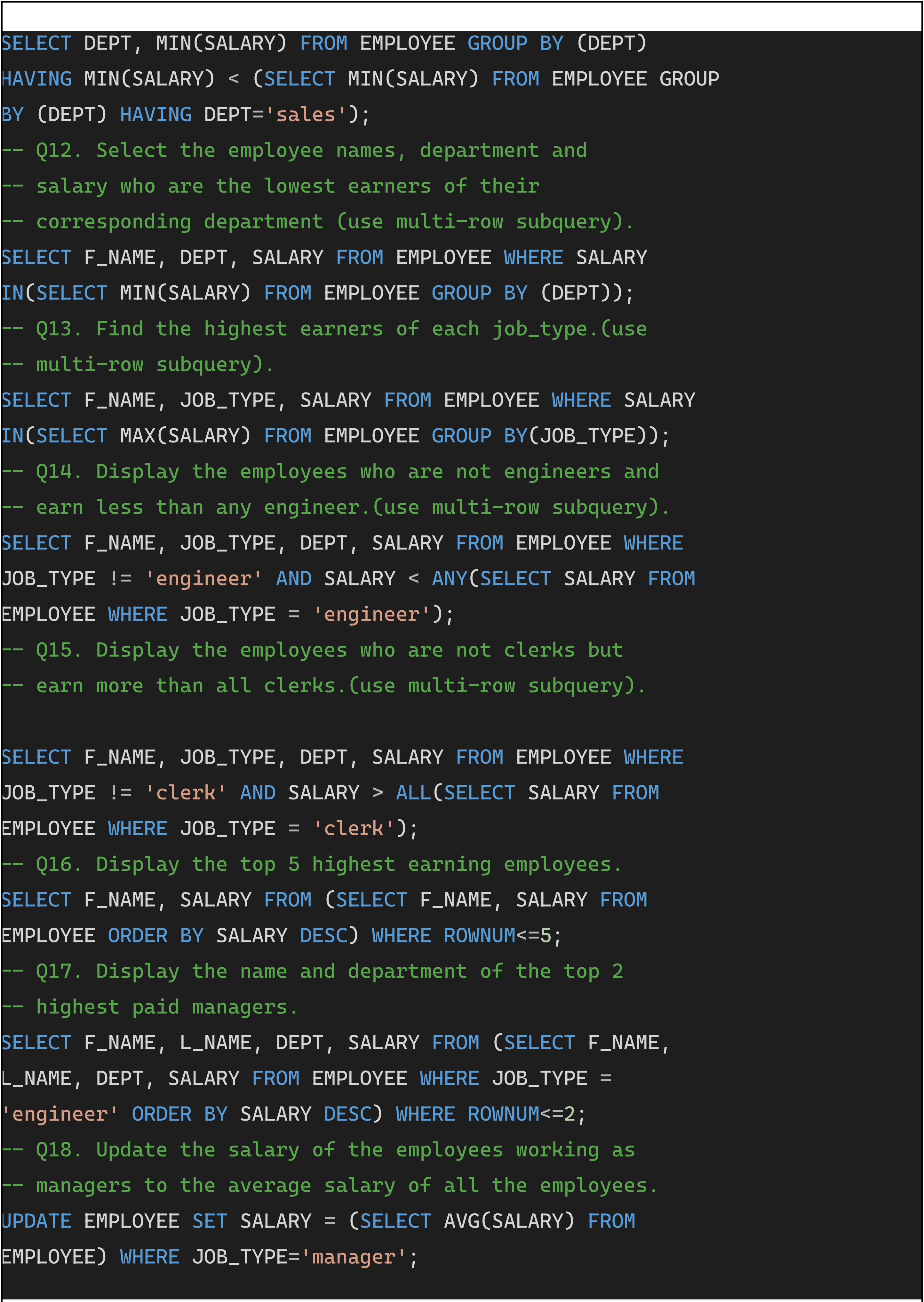
```
'engineer','production');
INSERT INTO Job_History Values(2,'1998-02-09','2002-02-28',
'salesman','sales');
INSERT INTO Job_History Values(1,'2001-07-01','2010-12-31',
'manager','rnd');
INSERT INTO Job_History Values(4,'2001-12-27','2016-09-19',
'sales_executive','marketing');
INSERT INTO Job_History Values(2,'2002-03-01','2015-03-30',
'sales_executive','marketing');
INSERT INTO Job_History Values(2,'2016-04-01','2017-12-16',
'manager','sales');
INSERT INTO Job_History Values(4,'2016-09-20','2017-12-16',
'asst. manager','sales');
INSERT INTO Job_History Values(6,'2000-07-16','2006-11-30',
'clerk','accounts');
INSERT INTO Job_History Values(5,'2002-03-20','2011-08-12',
'engineer','rnd');
INSERT INTO Job_History Values(1,'2011-01-01','2012-01-31',
'engineer','production');
SELECT * FROM Job_History;

CREATE TABLE Employee(
emp_id INT,
f_name VARCHAR(50),
l_name VARCHAR(50),
job_type VARCHAR(50),
salary INT,
commision INT,
dept VARCHAR(50),
manager_id INT,
DOJ DATE);
INSERT INTO Employee (emp_id,f_name,l_name,job_type,salary,dept,DOJ)
VALUES (1,'arun','khan','manager',90000,'production','1998-01-04');
INSERT INTO Employee (emp_id,f_name,l_name,job_type,salary,dept,DOJ)
VALUES (2,'barun','kumar','manager',80000,'marketing','1998-02-09');
INSERT INTO Employee
(emp_id,f_name,l_name,job_type,salary,dept,manager_id,DOJ)
```



```
VALUES (3,'chitra','kapoor','engineer',60000,'production',1,'1998-01-08');
INSERT INTO Employee
(emp_id,f_name,l_name,job_type,salary,dept,manager_id,DOJ)
VALUES (4,'dhirag','mishra','manager',75000,'sales',2,'2001-12-27');
INSERT INTO Employee
(emp_id,f_name,l_name,job_type,salary,dept,manager_id,DOJ)
VALUES (5,'emma','dutt','engineer',55000,'production',1,'2002-03-20');
INSERT INTO Employee (emp_id,f_name,l_name,job_type,salary,dept,DOJ)
VALUES (6,'floki','dutt','accountant',70000,'accounts','2000-07-16');
INSERT INTO Employee
(emp_id,f_name,l_name,job_type,salary,dept,manager_id,DOJ)
VALUES (7,'dheeraj','kumar','clerk',40000,'accounts',6,'2016-07-01');
INSERT INTO Employee (emp_id,f_name,l_name,job_type,salary,dept,DOJ)
VALUES (8,'saul','good','engineer',60000,'rnd','2014-09-06');
INSERT INTO Employee
(emp_id,f_name,l_name,job_type,salary,dept,manager_id,DOJ)
VALUES (9,'mou','bhat','clerk',30000,'sales',4,'2018-03-08');
INSERT INTO Employee
(emp_id,f_name,l_name,job_type,salary,commision,dept,manager_id,DOJ)
VALUES (10,'sunny','deol','salesman',20000,10000,'marketing',2,'2001-03-
31');
INSERT INTO Employee
(emp_id,f_name,l_name,job_type,salary,dept,manager_id,DOJ)
VALUES (11,'bobby','deol','engineer',35000,'rnd',8,'2017-10-17');
INSERT INTO Employee
(emp_id,f_name,l_name,job_type,salary,commision,dept,manager_id,DOJ)
VALUES (12,'amir','khan','salesman',15000,5000,'marketing',2,'2013-01-11');
SELECT * FROM Employee;
-- Q2. Display the emp_id along with every job_type
-- they have worked (including their current job_type).
-- (use union/union all)
SELECT EMP_ID, DEPT FROM EMPLOYEE UNION SELECT Emp_id,
Job_Type FROM Job_History;

-- Q3. Display the emp_id, d_name, and job_types
-- current and previous (if any) of all employees.(use
-- union/union all)
```

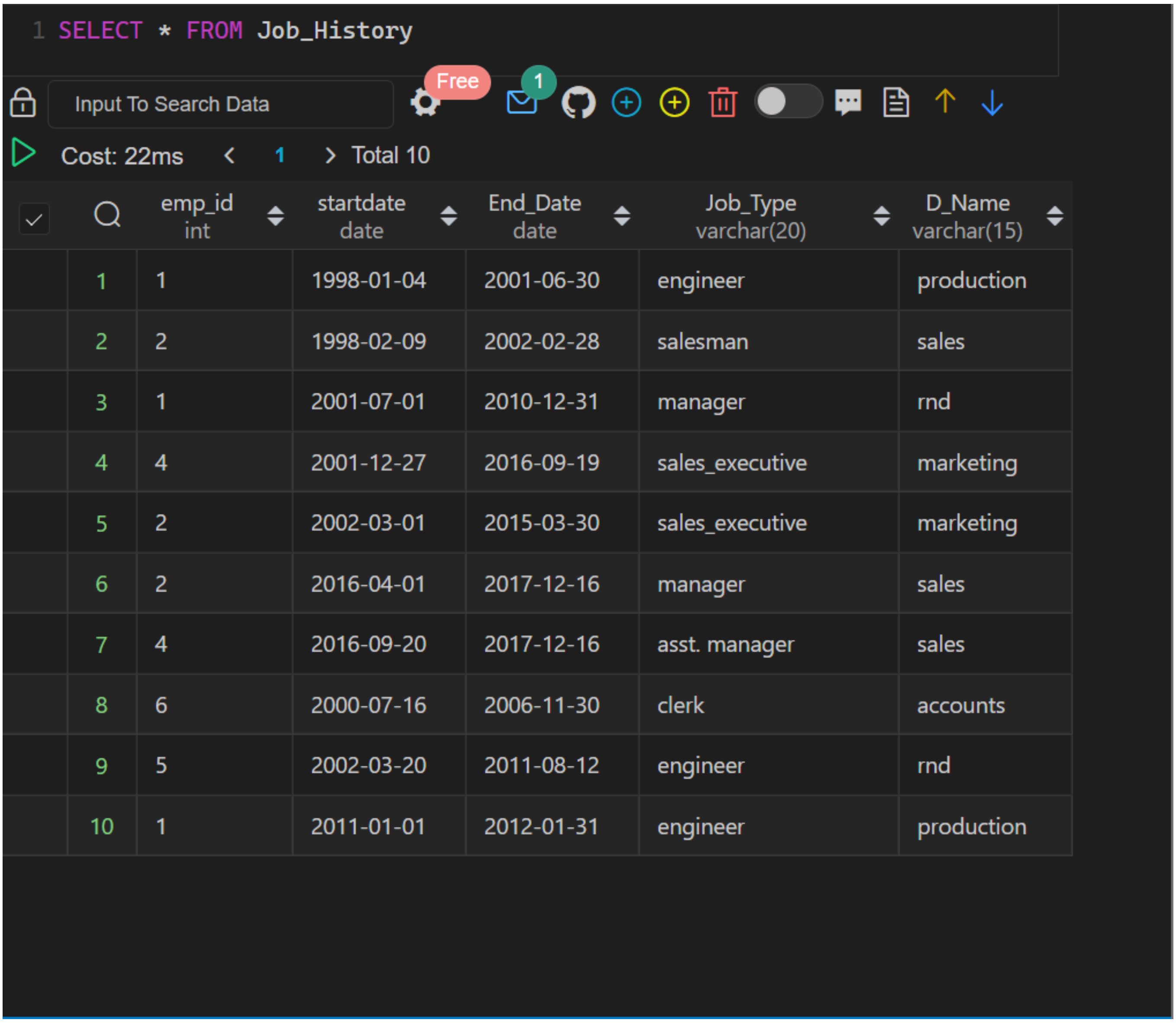


```
SELECT EMP_ID, JOB_TYPE, DEPT FROM EMPLOYEE UNION SELECT
Emp_id, Job_Type, D_Name FROM Job_History;
-- Q4. Display the emp_id and the job_type of
-- employees who currently have a job title that they
-- held previously.(use intersect)
SELECT EMP_ID, JOB_TYPE FROM EMPLOYEE INTERSECT SELECT
Emp_id, Job_Type FROM Job_History;
-- Q5. Find the employees who have not changed their
-- job for once.(use minus)
SELECT EMP_ID, JOB_TYPE FROM EMPLOYEE MINUS SELECT Emp_id,
Job_Type FROM Job_History;
-- Q6. Find the employees who earn more than Chitra.
-- (use single-row subquery)
SELECT EMP_ID FROM(SELECT EMP_ID FROM EMPLOYEE WHERE
SALARY>60000);
-- Q7. Find the employees details who have the same
-- job_type as of emp_id 7. (use single-row subquery)
SELECT * FROM (SELECT * FROM EMPLOYEE WHERE JOB_TYPE =
'clerk');
-- Q8.Display the employee names whose job is the same
-- as employee 3 and earn more than employee 7. (use
-- single-row subquery)
SELECT F_NAME FROM (SELECT F_NAME FROM EMPLOYEE WHERE
JOB_TYPE='engineer' AND SALARY>40000);
-- Q9. Display the employees earning less than the
-- average salary. (use single-row subquery)
SELECT F_NAME, SALARY FROM EMPLOYEE WHERE SALARY < (SELECT
AVG(SALARY) FROM EMPLOYEE);
-- Q10. Find the job_type with the lowest average

salary. (use single-row subquery)
SELECT JOB_TYPE FROM (SELECT JOB_TYPE, AVG(SALARY) FROM
EMPLOYEE GROUP BY JOB_TYPE HAVING AVG(SALARY) = (SELECT
MIN(AVG(SALARY)) FROM EMPLOYEE GROUP BY(JOB_TYPE)));
-- Q11. Display all the department names whose minimum
-- salary is greater than the minimum salary of the Sales
-- department.
```



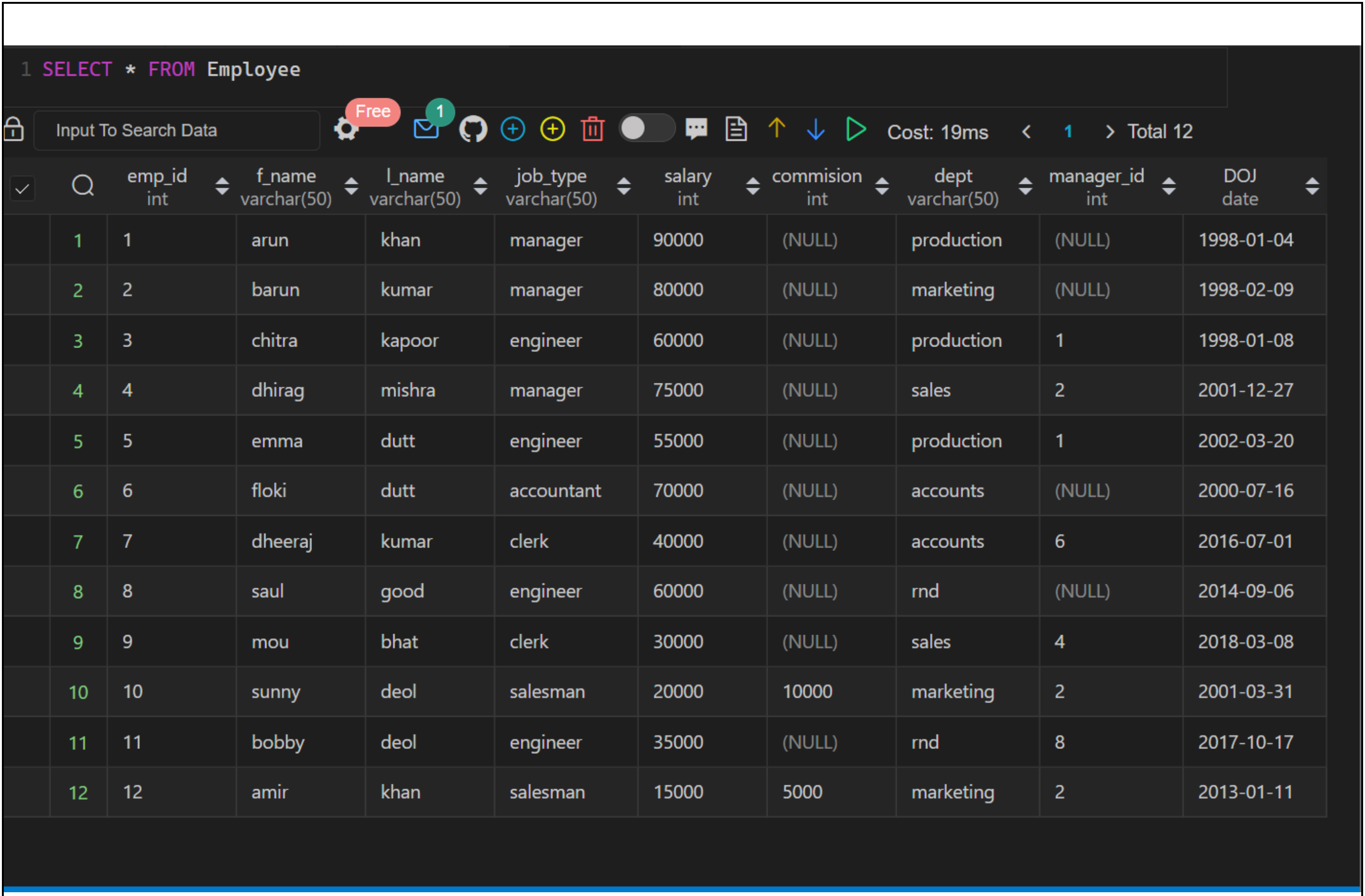
```
SELECT DEPT, MIN(SALARY) FROM EMPLOYEE GROUP BY (DEPT)
HAVING MIN(SALARY) < (SELECT MIN(SALARY) FROM EMPLOYEE GROUP
BY (DEPT) HAVING DEPT='sales');
-- Q12. Select the employee names, department and
-- salary who are the lowest earners of their
-- corresponding department (use multi-row subquery).
SELECT F_NAME, DEPT, SALARY FROM EMPLOYEE WHERE SALARY
IN(SELECT MIN(SALARY) FROM EMPLOYEE GROUP BY (DEPT));
-- Q13. Find the highest earners of each job_type.(use
-- multi-row subquery).
SELECT F_NAME, JOB_TYPE, SALARY FROM EMPLOYEE WHERE SALARY
IN(SELECT MAX(SALARY) FROM EMPLOYEE GROUP BY(JOB_TYPE));
-- Q14. Display the employees who are not engineers and
-- earn less than any engineer.(use multi-row subquery).
SELECT F_NAME, JOB_TYPE, DEPT, SALARY FROM EMPLOYEE WHERE
JOB_TYPE != 'engineer' AND SALARY < ANY(SELECT SALARY FROM
EMPLOYEE WHERE JOB_TYPE = 'engineer');
-- Q15. Display the employees who are not clerks but
-- earn more than all clerks.(use multi-row subquery).

SELECT F_NAME, JOB_TYPE, DEPT, SALARY FROM EMPLOYEE WHERE
JOB_TYPE != 'clerk' AND SALARY > ALL(SELECT SALARY FROM
EMPLOYEE WHERE JOB_TYPE = 'clerk');
-- Q16. Display the top 5 highest earning employees.
SELECT F_NAME, SALARY FROM (SELECT F_NAME, SALARY FROM
EMPLOYEE ORDER BY SALARY DESC) WHERE ROWNUM<=5;
-- Q17. Display the name and department of the top 2
-- highest paid managers.
SELECT F_NAME, L_NAME, DEPT, SALARY FROM (SELECT F_NAME,
L_NAME, DEPT, SALARY FROM EMPLOYEE WHERE JOB_TYPE =
'engineer' ORDER BY SALARY DESC) WHERE ROWNUM<=2;
-- Q18. Update the salary of the employees working as
-- managers to the average salary of all the employees.
UPDATE EMPLOYEE SET SALARY = (SELECT AVG(SALARY) FROM
EMPLOYEE) WHERE JOB_TYPE='manager';
```

OUTPUTS:-



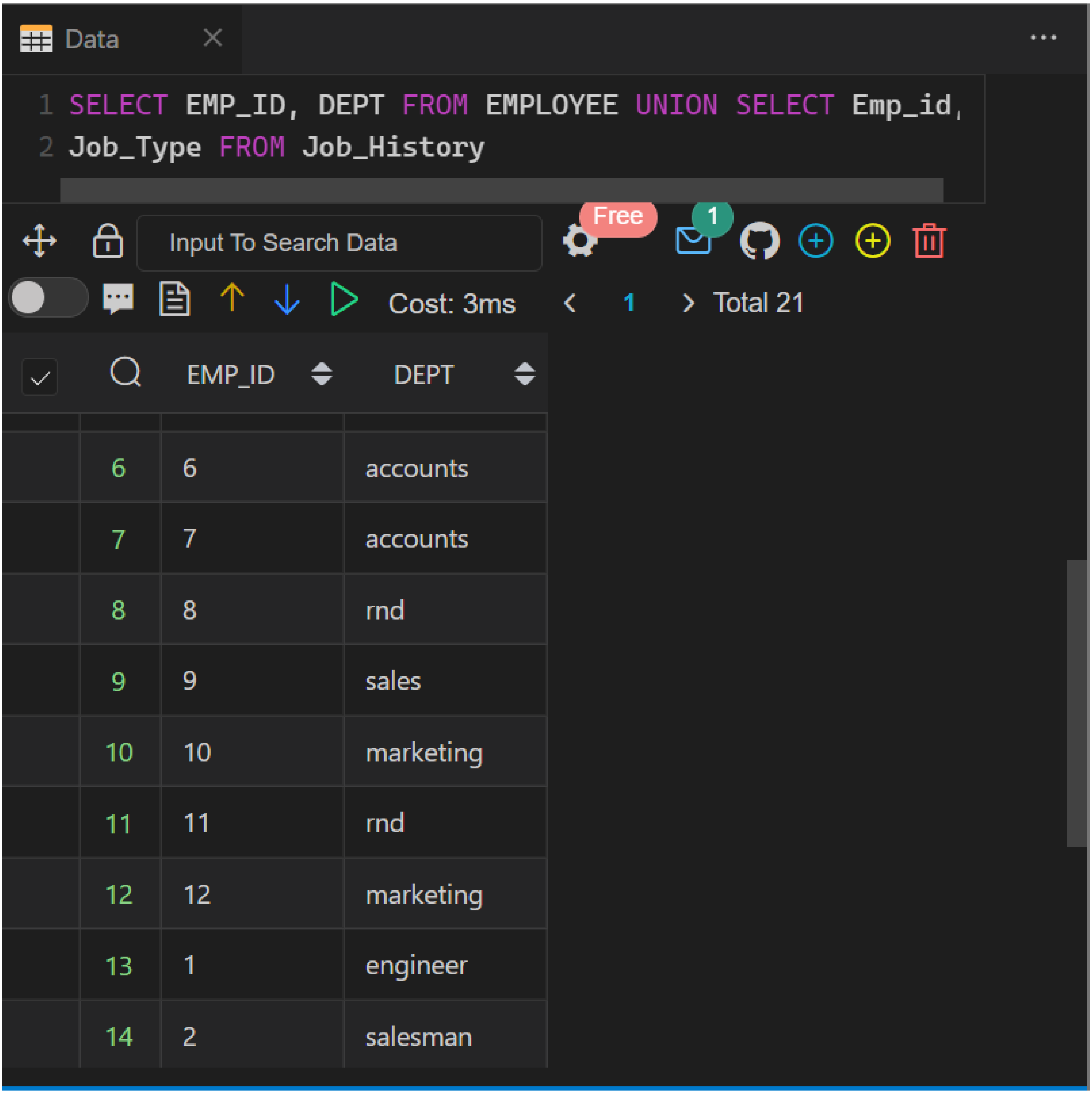
```
SELECT * FROM Job_History
```

	emp_id int	startdate date	End_Date date	Job_Type varchar(20)	D_Name varchar(15)
1	1	1998-01-04	2001-06-30	engineer	production
2	2	1998-02-09	2002-02-28	salesman	sales
3	1	2001-07-01	2010-12-31	manager	rnd
4	4	2001-12-27	2016-09-19	sales_executive	marketing
5	2	2002-03-01	2015-03-30	sales_executive	marketing
6	2	2016-04-01	2017-12-16	manager	sales
7	4	2016-09-20	2017-12-16	asst. manager	sales
8	6	2000-07-16	2006-11-30	clerk	accounts
9	5	2002-03-20	2011-08-12	engineer	rnd
10	1	2011-01-01	2012-01-31	engineer	production



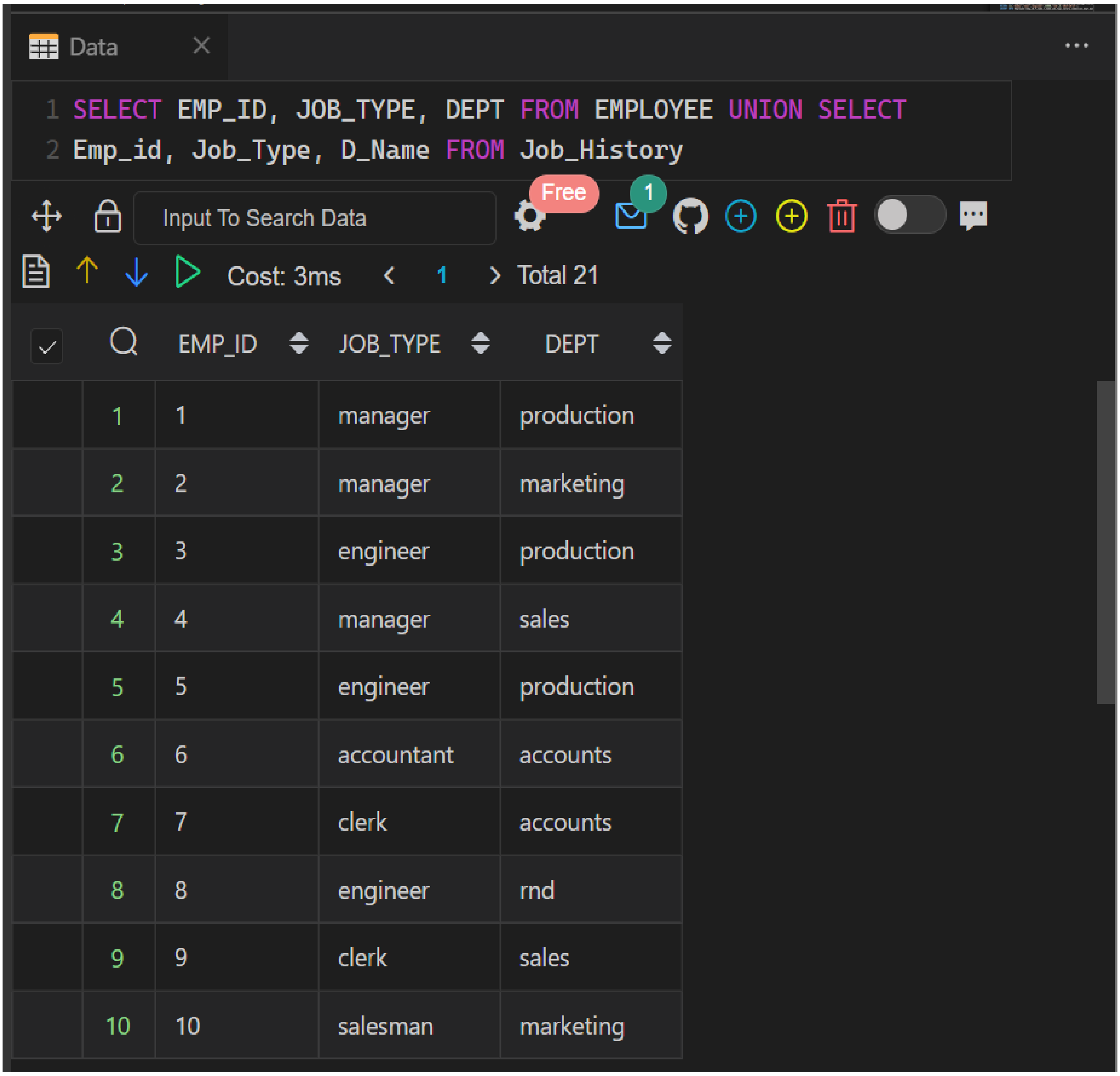
```sql
SELECT * FROM Employee
```

	emp_id int	f_name varchar(50)	l_name varchar(50)	job_type varchar(50)	salary int	commision int	dept varchar(50)	manager_id int	DOJ date
1	1	arun	khan	manager	90000	(NULL)	production	(NULL)	1998-01-04
2	2	barun	kumar	manager	80000	(NULL)	marketing	(NULL)	1998-02-09
3	3	chitra	kapoor	engineer	60000	(NULL)	production	1	1998-01-08
4	4	dhirag	mishra	manager	75000	(NULL)	sales	2	2001-12-27
5	5	emma	dutt	engineer	55000	(NULL)	production	1	2002-03-20
6	6	floki	dutt	accountant	70000	(NULL)	accounts	(NULL)	2000-07-16
7	7	dheeraj	kumar	clerk	40000	(NULL)	accounts	6	2016-07-01
8	8	saul	good	engineer	60000	(NULL)	rnd	(NULL)	2014-09-06
9	9	mou	bhat	clerk	30000	(NULL)	sales	4	2018-03-08
10	10	sunny	deol	salesman	20000	10000	marketing	2	2001-03-31
11	11	bobby	deol	engineer	35000	(NULL)	rnd	8	2017-10-17
12	12	amir	khan	salesman	15000	5000	marketing	2	2013-01-11



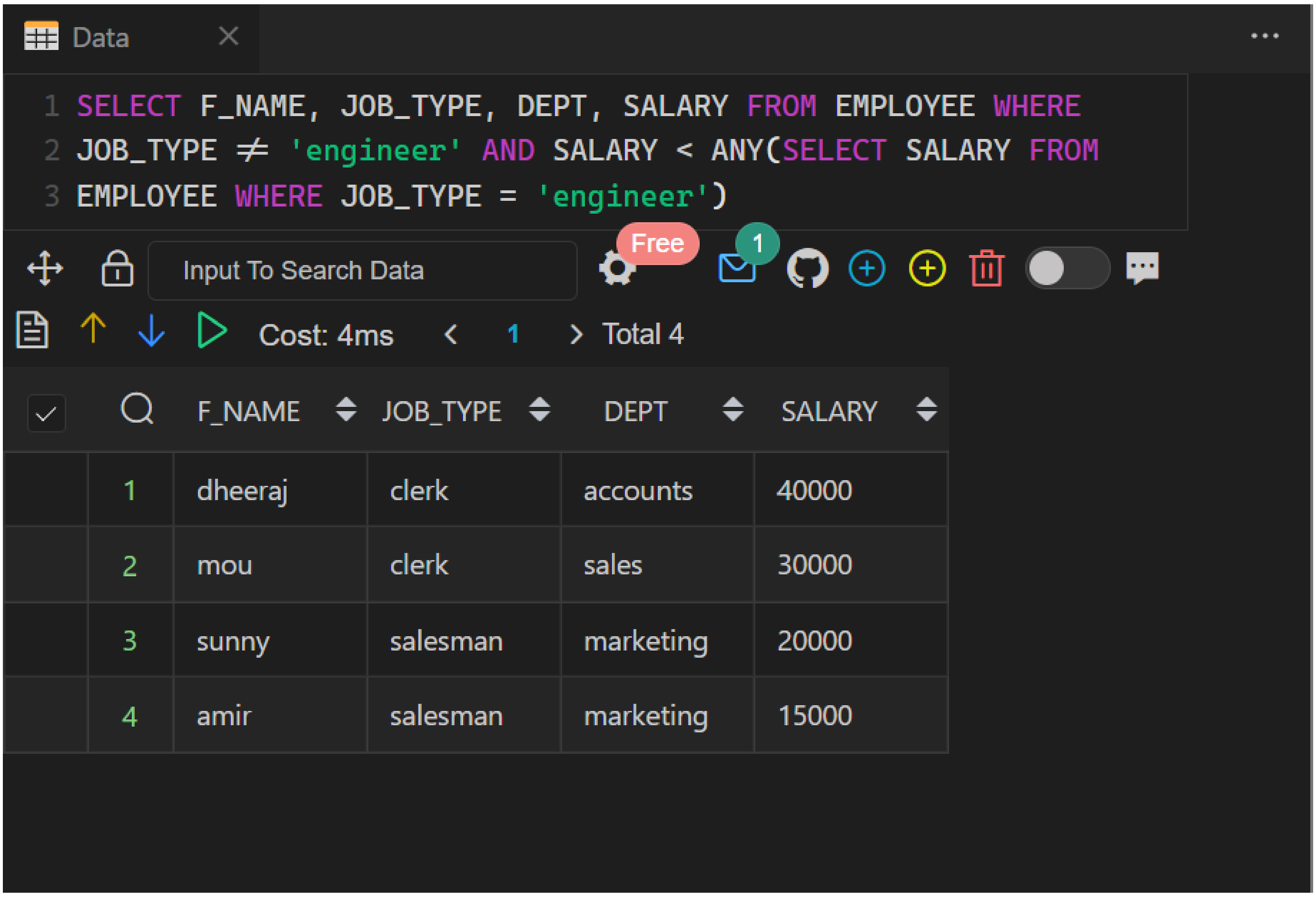
```
SELECT EMP_ID, DEPT FROM EMPLOYEE UNION SELECT Emp_id,
Job_Type FROM Job_History
```

	EMP_ID	DEPT
6	6	accounts
7	7	accounts
8	8	rnd
9	9	sales
10	10	marketing
11	11	rnd
12	12	marketing
13	1	engineer
14	2	salesman



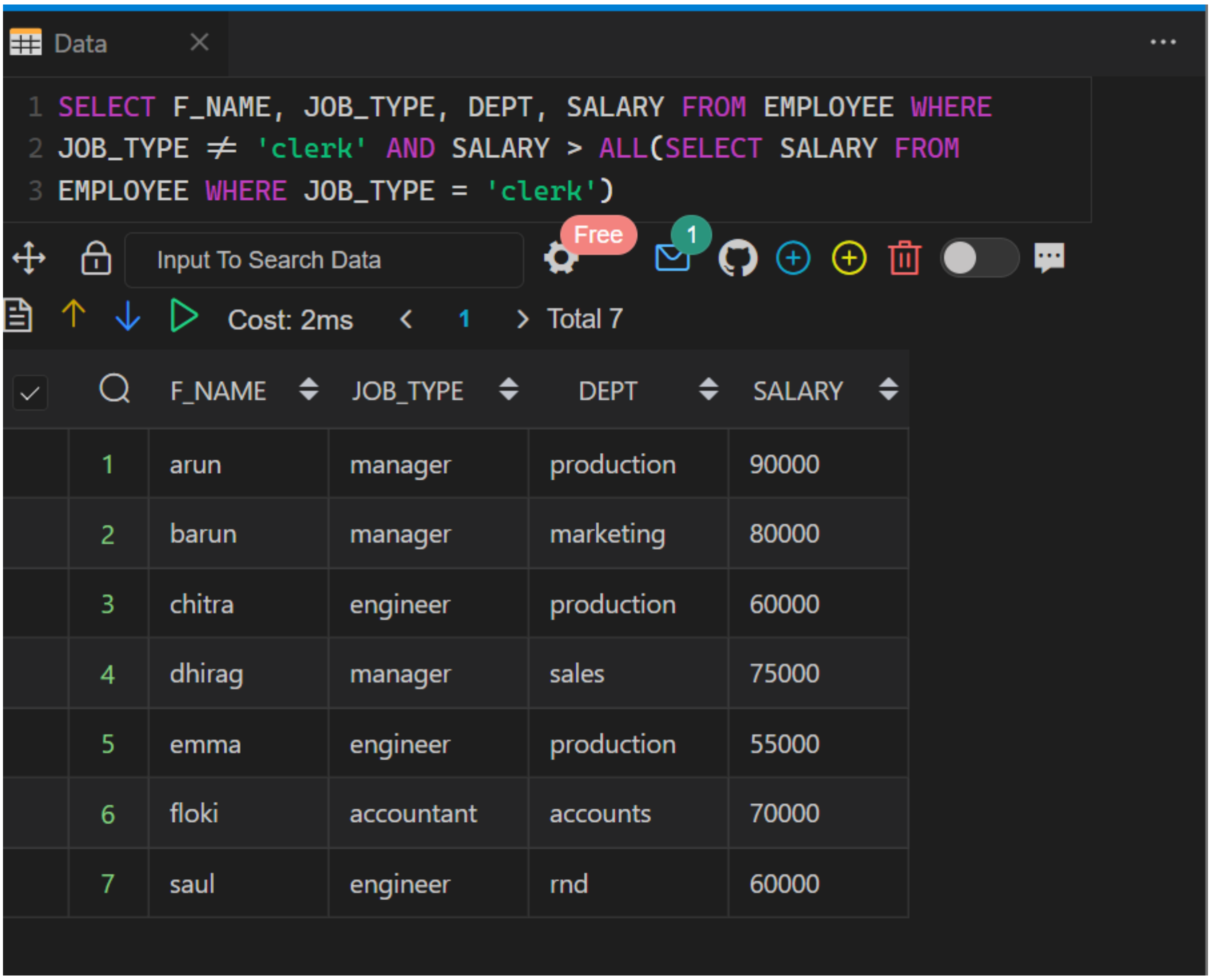
```sql
SELECT EMP_ID, JOB_TYPE, DEPT FROM EMPLOYEE UNION SELECT
Emp_id, Job_Type, D_Name FROM Job_History
```

	EMP_ID	JOB_TYPE	DEPT
1	1	manager	production
2	2	manager	marketing
3	3	engineer	production
4	4	manager	sales
5	5	engineer	production
6	6	accountant	accounts
7	7	clerk	accounts
8	8	engineer	rnd
9	9	clerk	sales
10	10	salesman	marketing



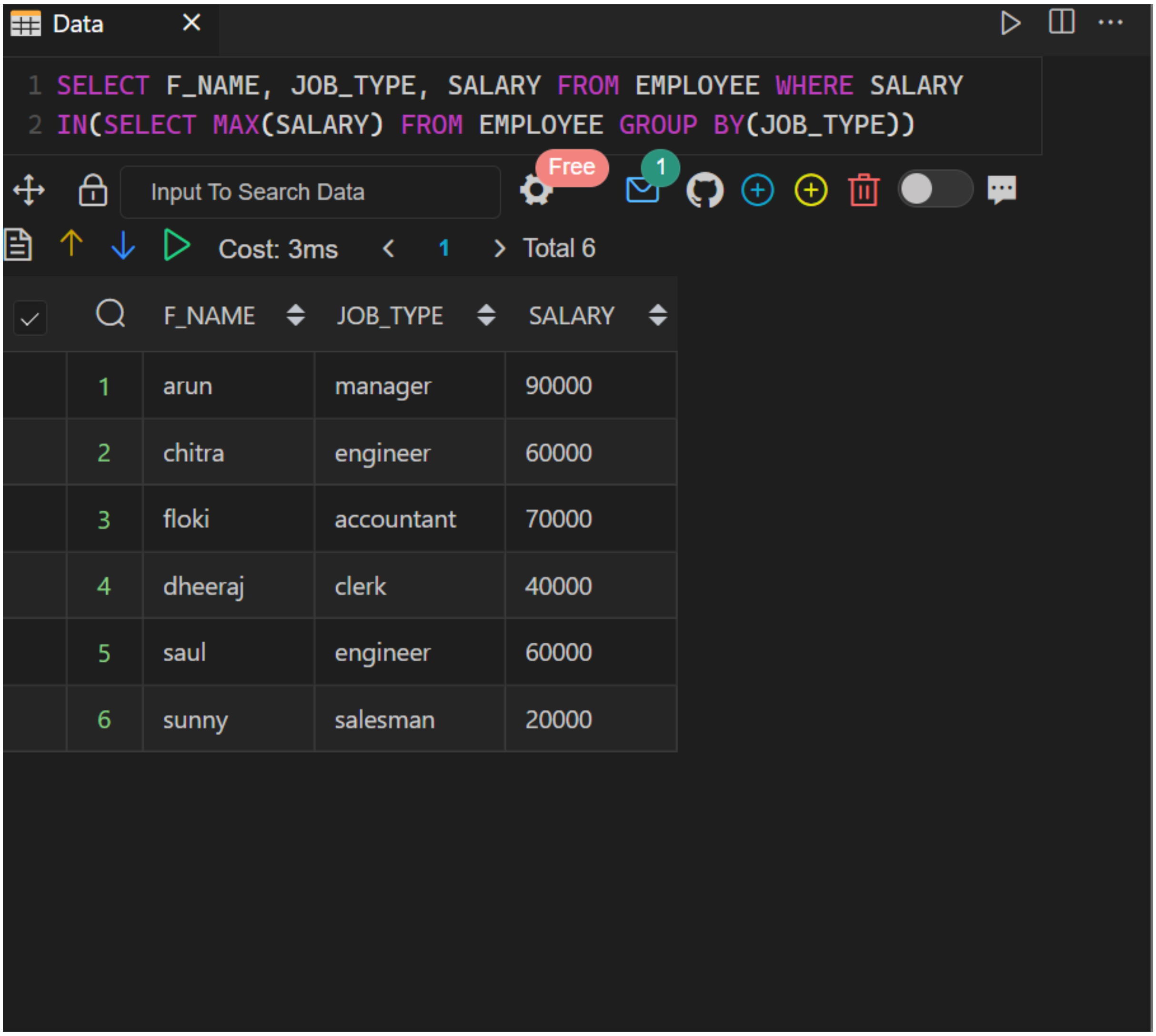
```sql
SELECT F_NAME, JOB_TYPE, DEPT, SALARY FROM EMPLOYEE WHERE
JOB_TYPE ≠ 'engineer' AND SALARY < ANY(SELECT SALARY FROM
EMPLOYEE WHERE JOB_TYPE = 'engineer')
```

	F_NAME	JOB_TYPE	DEPT	SALARY
1	dheeraj	clerk	accounts	40000
2	mou	clerk	sales	30000
3	sunny	salesman	marketing	20000
4	amir	salesman	marketing	15000



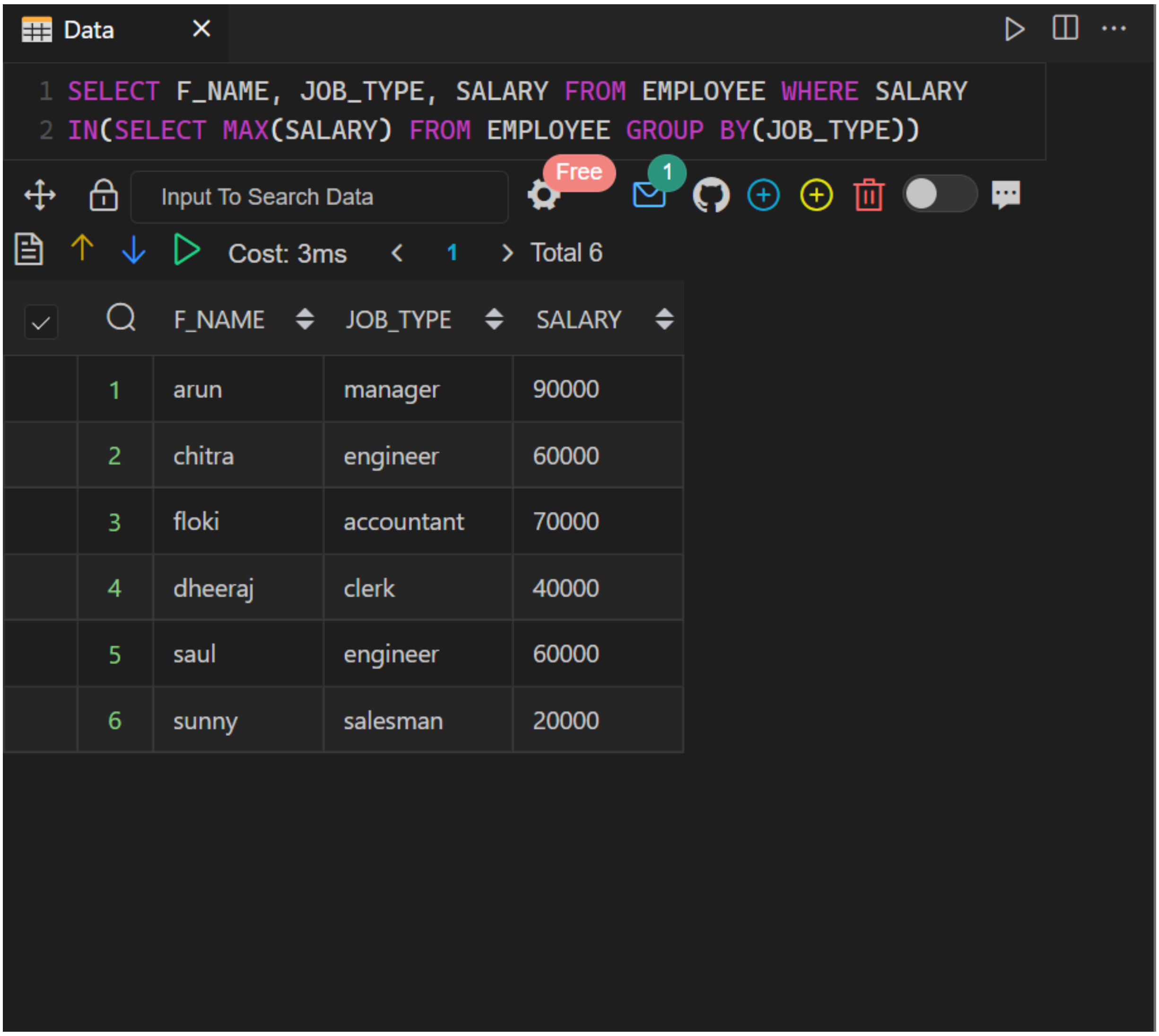
```sql
SELECT F_NAME, JOB_TYPE, DEPT, SALARY FROM EMPLOYEE WHERE
JOB_TYPE ≠ 'clerk' AND SALARY > ALL(SELECT SALARY FROM
EMPLOYEE WHERE JOB_TYPE = 'clerk')
```

	F_NAME	JOB_TYPE	DEPT	SALARY
1	arun	manager	production	90000
2	barun	manager	marketing	80000
3	chitra	engineer	production	60000
4	dhirag	manager	sales	75000
5	emma	engineer	production	55000
6	floki	accountant	accounts	70000
7	saul	engineer	rnd	60000



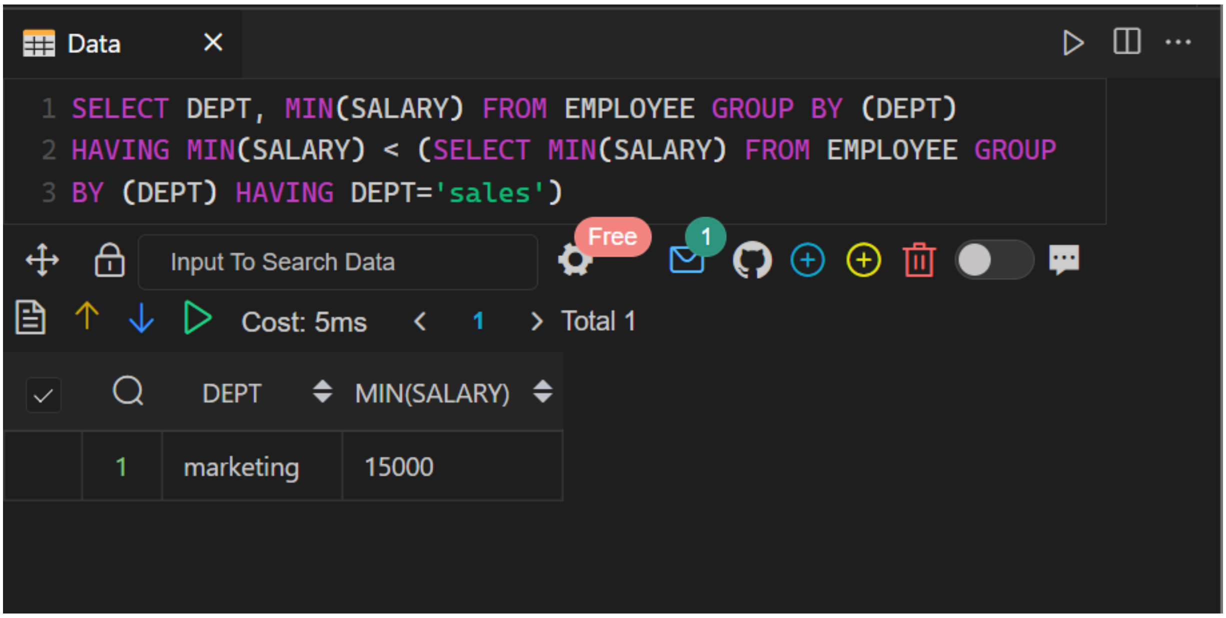
```
SELECT F_NAME, JOB_TYPE, SALARY FROM EMPLOYEE WHERE SALARY
IN(SELECT MAX(SALARY) FROM EMPLOYEE GROUP BY(JOB_TYPE))
```

	F_NAME	JOB_TYPE	SALARY
1	arun	manager	90000
2	chitra	engineer	60000
3	floki	accountant	70000
4	dheeraj	clerk	40000
5	saul	engineer	60000
6	sunny	salesman	20000



```
SELECT F_NAME, JOB_TYPE, SALARY FROM EMPLOYEE WHERE SALARY
IN(SELECT MAX(SALARY) FROM EMPLOYEE GROUP BY(JOB_TYPE))
```

	F_NAME	JOB_TYPE	SALARY
1	arun	manager	90000
2	chitra	engineer	60000
3	floki	accountant	70000
4	dheeraj	clerk	40000
5	saul	engineer	60000
6	sunny	salesman	20000



```
SELECT DEPT, MIN(SALARY) FROM EMPLOYEE GROUP BY (DEPT)
HAVING MIN(SALARY) < (SELECT MIN(SALARY) FROM EMPLOYEE GROUP
BY (DEPT) HAVING DEPT='sales')
```

	DEPT	MIN(SALARY)
1	marketing	15000