

Database Management System (MCA 291)

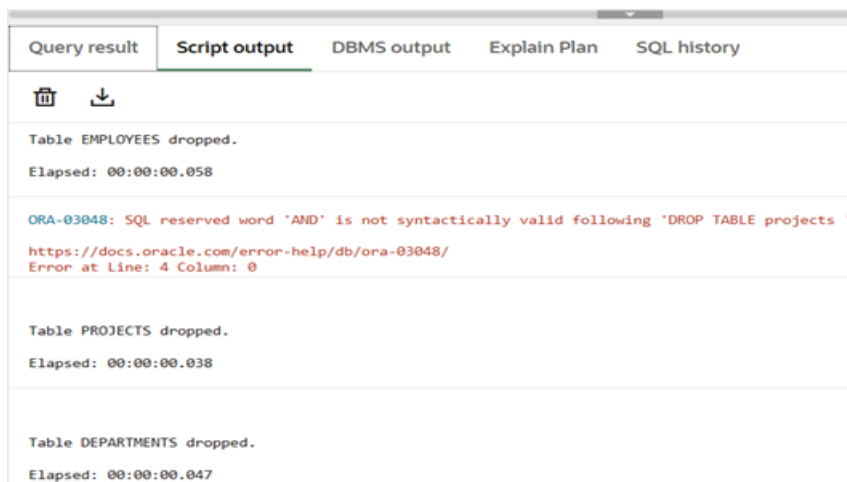
Assignment - 2

2. Implement all the table with create table query and then drop those 3 tables along with the constraints.

=> `DROP TABLE departments CASCADE constraints;`

`DROP TABLE projects CASCADE constraints`

`DROP TABLE employees CASCADE constraints`

Query result	Script output	DBMS output	Explain Plan	SQL history
				

3. Create these 3 tables first then add the constraint with alter table query.

=> `alter table Departments add CONSTRAINT
Department_pk PRIMARY key (DepartmentId)`

`alter table orders add foreign key (DepartmentId)
references Departments(DepartmentId)`

`alter table Employees orders add foreign key
(DepartmentId) references Departments(DepartmentId)
alter table Employees add foreign key (DepartmentId)
references Departments(DepartmentId)`

`alter table Projects add foreign key (EmployeeId)
references Projects(EmployeeId)`

`alter table Projctcs add primary key (ProjectId)
Alter table Projects add foreign key (Employeeid)
REFERENCES Employees(Employeeid)`

`Alter table Employees add foreign key (Departmentid)
REFERENCES Departments(Departmentid)`



```
Elapsed: 00:00:00.015  
5 rows selected.
```

```
Table PROJECTS altered.
```

```
Elapsed: 00:00:00.020
```

4. a. Find employees without an email address (if NULL values exist)

```
=> select Employeeid, firstname, lastname from  
EMPLOYEES where EMAIL is NULL
```

Query result	Script output	DBMS output	Explain Plan	SQL history
 				
0 rows selected.				

b. Find employees who have an email address registered

```
=> select Employeeid, firstname, lastname from  
EMPLOYEES where EMAIL is not NULL
```

Query result		Script output
 		
EMPLOYEEID	FIRSTNAME	LASTNAME
-----	-----	-----
1	John	Doe
2	Alice	Smith
3	Bob	Johnson
4	Emma	Davis
5	Charlie	Brown
Elapsed: 00:00:00.004		
5 rows selected.		

c. Find employees who were hired in June of any year

=> `SELECT *FROM employees WHERE EXTRACT (MONTH FROM HIREDATE) = 6`



Query result

Script output

DBMS output

Explain Plan

SQL history



Elapsed: 00:00:00.005

1 rows selected.

EMPLOYEEID FIRSTNAME LASTNAME

4

Emma

Davis

Elapsed: 00:00:00.003

1 rows selected.

d. Find employees hired in the last 5 years

=> `select Employeeid, firstname, lastname from EMPLOYEES where HIREDATE >= ADD_MONTHS (SYSDATE, -60)`

Query result

Script output

DBMS output

Explain Plan

SQL history

🗑️

📄

Elapsed: 00:00:00.004

2 rows selected.

EMPLOYEEID	FIRSTNAME	LASTNAME
1	John	Doe
4	Emma	Davis

Elapsed: 00:00:00.004

2 rows selected.

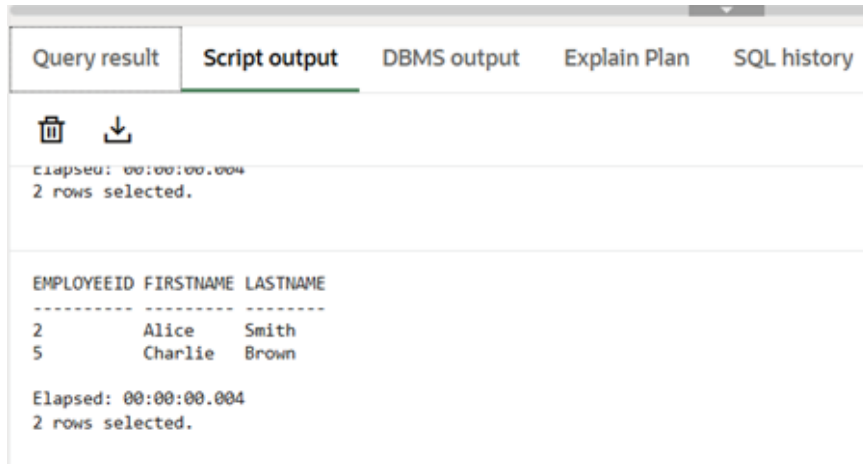
[I Notices](#)
[Terms and Conditions](#)
[Your Privacy Rights](#)
[Delete Your Live SQL Account](#)
[Cookie Preference](#)

[Affiliates](#)

All rights reserved.

e. Find all employees who belong to the IT department (assuming DepartmentID = 2)

=> `select Employeeid, firstname, lastname from EMPLOYEES where DEPARTMENTID = 2`

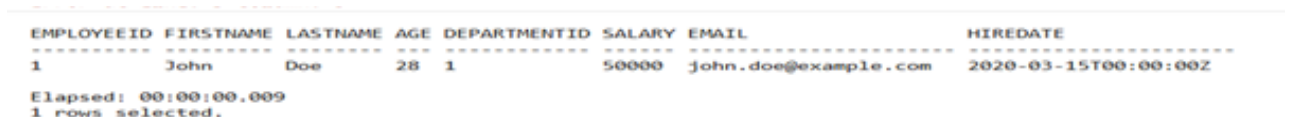


The screenshot shows a database query result window with tabs for 'Query result', 'Script output', 'DBMS output', 'Explain Plan', and 'SQL history'. The 'Query result' tab is active. It displays a table with columns EMPLOYEEID, FIRSTNAME, and LASTNAME. The results show two rows: Alice Smith (EmployeeID 2) and Charlie Brown (EmployeeID 5). The window also shows the elapsed time as 00:00:00.004 and 2 rows selected.

EMPLOYEEID	FIRSTNAME	LASTNAME
2	Alice	Smith
5	Charlie	Brown

f. Find employees in the HR department by department name

=> `SELECT e.*
FROM employees e
JOIN departments d
ON e.departmentid = d.departmentid
WHERE d.departmentname = 'HR'`



The screenshot shows a database query result window displaying a table with columns EMPLOYEEID, FIRSTNAME, LASTNAME, AGE, DEPARTMENTID, SALARY, EMAIL, and HIREDATE. The results show one row: John Doe (EmployeeID 1, Age 28, DepartmentID 1, Salary 50000, Email john.doe@example.com, HireDate 2020-03-15T00:00:00Z). The window also shows the elapsed time as 00:00:00.009 and 1 row selected.

EMPLOYEEID	FIRSTNAME	LASTNAME	AGE	DEPARTMENTID	SALARY	EMAIL	HIREDATE
1	John	Doe	28	1	50000	john.doe@example.com	2020-03-15T00:00:00Z

g. Find employees who are not assigned to any projects

=> `select employeeid from projects WHERE PROJECTNAME is NULL`

Query result

Script output

DBMS output

Explain Plan

SQL history

EMPLOYEEID	FIRSTNAME	LASTNAME	AGE	DEPARTMENTID	SALARY	EMAIL	HIREDATE
1	John	Doe	28	1	50000	john.doe@example.com	2020-03-15T00:00:00Z

Elapsed: 00:00:00.009

1 rows selected.

0 rows selected.

h. Count the number of employees in each department

```
=> SELECT departmentid, COUNT(*) AS employee_count
FROM employees
GROUP BY departmentid
```

Query result	Script output										
<div> </div> <table> <thead> <tr> <th>DEPARTMENTID</th><th>EMPLOYEE_COUNT</th></tr> </thead> <tbody> <tr> <td>1</td><td>1</td></tr> <tr> <td>2</td><td>2</td></tr> <tr> <td>3</td><td>1</td></tr> <tr> <td>4</td><td>1</td></tr> </tbody> </table> <p>Elapsed: 00:00:00.002 4 rows selected.</p>		DEPARTMENTID	EMPLOYEE_COUNT	1	1	2	2	3	1	4	1
DEPARTMENTID	EMPLOYEE_COUNT										
1	1										
2	2										
3	1										
4	1										

i. Count the number of projects handled by each employee

```
=> SELECT EMPLOYEEID, COUNT(*) AS project_count
FROM PROJECTS
GROUP BY EMPLOYEEID
```

EMPLOYEEID	PROJECT_COUNT
2	1
3	1
4	1
1	1
5	1

Elapsed: 00:00:00.003
5 rows selected.

j. Get a list of all unique department IDs in the Employees table

=> `SELECT DISTINCT departmentid FROM employees`

