

## EXERCISE 11 CREATING VIEWS

NAME: VIDHYASHREE J

REG NO:241801310

1. Create a view called EMPLOYEE\_VU based on the employee numbers, employee names and department numbers from the EMPLOYEES table. Change the heading for the employee name to EMPLOYEE.

ORACLE® Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
CREATE OR REPLACE VIEW employee_vu (employee_number, employee, department_number)
AS SELECT employee_id, first_name || ' ' || last_name, department_id
FROM employees;
```

Results Explain Describe Saved SQL History

View created.  
0.04 seconds

2. Display the contents of the EMPLOYEES\_VU view.

## ORACLE Database Express Edition

User: SYSTEM

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
SELECT * FROM employee_vu;
```

**Results** Explain Describe Saved SQL History

EMPLOYEE_NUMBER	EMPLOYEE	DEPARTMENT_NUMBER
100	Steven King	90
101	Neena Kochhar	90
102	Lex De Haan	90

3 rows returned in 0.00 seconds [CSV Export](#)

3. Select the view name and text from the USER\_VIEWS data dictionary views.

## ORACLE® Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT view_name, text
FROM user_views
WHERE view_name = 'EMPLOYEE_VU';
```

Results Explain Describe Saved SQL History

VIEW_NAME	TEXT
EMPLOYEE_VU	SELECT employee_id, first_name    ''    last_name, department_id FROM employees

1 rows returned in 0.00 seconds [CSV Export](#)

- Using your EMPLOYEES\_VU view, enter a query to display all employees names and department.

# ORACLE® Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 ▾  
SELECT employee, department\_number  
FROM employee\_vu;

Results Explain Describe Saved SQL History

EMPLOYEE	DEPARTMENT_NUMBER
Steven King	90
Neena Kochhar	90
Lex De Haan	90

3 rows returned in 0.00 seconds [CSV Export](#)

5. Create a view named DEPT50 that contains the employee number, employee last names and department numbers for all employees in department 50. Label the view columns EMPNO, EMPLOYEE and DEPTNO. Do not allow an employee to be reassigned to another department through the view.

# ORACLE® Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
CREATE OR REPLACE VIEW dept50 (empno, employee, deptno)
AS SELECT employee_id, last_name, department_id
FROM employees
WHERE department_id = 50
WITH CHECK OPTION CONSTRAINT dept50_ck;
```

Results Explain Describe Saved SQL History

View created.

0.02 seconds

6. Display the structure and contents of the DEPT50 view.

## ORACLE® Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
DESCRIBE dept50;
```

Results Explain Describe Saved SQL History

Object Type **VIEW** Object DEPT50

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT50	EMPNO	Number	-	-	0	-	✓	-	-
	EMPLOYEE	Varchar2	50	-	-	-	✓	-	-
	DEPTNO	Number	-	-	0	-	✓	-	-
1 - 3									

## ORACLE® Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT * FROM dept50;
```

Results Explain Describe Saved SQL History

EMPNO	EMPLOYEE	DEPTNO
999	Matos	50

1 rows returned in 0.00 seconds

[CSV Export](#)

7. Attempt to reassign Matos to department 80.

# ORACLE® Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
UPDATE employees
SET department_id = 80
WHERE last_name = 'Matos';
|
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.00 seconds

8. Create a view called SALARY\_VU based on the employee last names, department names, salaries, and salary grades for all employees. Use the Employees, DEPARTMENTS and JOB\_GRADE tables. Label the column Employee, Department, salary, and Grade respectively.

# ORACLE® Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
CREATE OR REPLACE VIEW salary_vu (employee, department, salary, grade)
AS SELECT e.last_name,
          d.department_name,
          e.salary,
          j.grade_level |
     FROM employees e
JOIN departments d ON e.department_id = d.department_id
JOIN job_grade j ON e.salary BETWEEN j.lowest_sal AND j.highest_sal;
```

Results Explain Describe Saved SQL History

View created.

0.02 seconds