

Join Tables Exam

Q1: Examine the data in the EMPLOYEES and DEPARTMENTS tables.

EMPLOYEES

LAST_NAME	DEPARTMENT_ID	SALARY
Getz	10	3000
Davis	20	1500
Bill	20	2200
Davis	30	5000
Kochhar		5000

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME
10	Sales
20	Marketing
30	Accounts
40	Administration

You want to retrieve all employees, whether or not they have matching departments in the departments table. Which query would you use?

- A. SELECT last_name, department_name FROM employees , epartments(+);
- B. SELECT last_name, department_name FROM employees JOIN departments(+);
- C. SELECT last_name, department_name ON (e. department_ id = d. departments_id); FROM employees(+) e JOIN departments d
- D. SELECT last_name, department_name FROM employees e RIGHT OUTER JOIN departments d ON (e.department_id = d.department_id);
- E. SELECT last_name, department_name FROM employees(+) , departments ON (e. department _ id = d. department _id);
- F. SELECT last_name, department_name FROM employees e LEFT OUTER JOIN departments d ON (e. department _ id = d. department _id);

Q2: Examine the structure of the EMPLOYEES and DEPARTMENTS tables.

Column	name	Data type	Remarks
EMPLOYEE_ID	NUMBER	NOT NULL, Primary Key	
EMP_NAME	VARCHAR2	(30)	
JOB_ID	VARCHAR2	(20)	
SALARY	NUMBER		
MGR_ID	NUMBER		
References	EMPLOYEE_ID	COLUMN	DEPARTMENT ID
	NUMBER		
Foreign key to DEPARTMENT ID column of the DEPARTMENTS table			

DEPARTMENTS

Column	name	Data type	Remarks
DEPARTMENT_ID	NUMBER	NOT NULL, Primary Key	
DEPARTMENT_NAME	VARCHAR2	(30)	
MGR_ID	NUMBER		References MGR_ID column of the EMPLOYEES table

Evaluate this SQL statement. SELECT employee_id, e.department_id, department_name, salary FROM employees e, departments d WHERE e. department_id = d.department_id;

Which SQL statement is equivalent to the above SQL statement?

- A.** SELECT employee_id, department_id, department_name, salary FROM employees WHERE department_id IN (SELECT department_id FROM departments);
- B.** SELECT employee_id, department_id, department_name, salary FROM employees NATURAL JOIN departments;
- C.** SELECT employee_id, d.department_id, department_name, salary FROM employees e JOIN departments d ON e.department _ id = d. department_id;
- D.** SELECT employee_id, department_id, department_name, Salary FROM employees JOIN departments USING (e.department_id, d.department_id);

Q3: Examine the structure of the EMPLOYEES, DEPARTMENTS, and LOCATIONS tables. EMPLOYEES

EMPLOYEE_ID	NUMBER	NOT NULL, Primary Key
EMP_NAME	VARCHAR2 (30)	
JOB_ID	VARCHAR2 (20)	
SALARY	NUMBER	
MGR_ID	NUMBER	References EMPLOYEE_ID column
DEPARTMENT_ID	NUMBER	Foreign key to DEPARTMENT_ID column of the DEPARTMENTS table

DEPARTMENTS

DEPARTMENT_ID	NUMBER	NOT NULL, Primary Key
DEPARTMENT_NAME	VARCHAR2 (30)	
MGR_ID	NUMBER	References MGR_ID column of the EMPLOYEES table
LOCATION_ID	NUMBER	Foreign key to LOCATION_ID column of the LOCATIONS table

LOCATIONS VARCHAR2

LOCATION_ID	NUMBER	NOT NULL, Primary Key
-------------	--------	-----------------------

CITY

[30] Which two SQL statements produce the name, department name, and the city of all the employees who earn more than 10000? (Choose two)

A. SELECT emp_name, department_name, city
FROM employees e
JOIN departments d
USING (department_id)
JOIN locations l
USING (location_id)
WHERE salary > 10000;

B. SELECT emp_name, department_name, city
FROM employees e, departments d, locations 1
JOIN ON (e.department_id = d.department_id)
AND (d.location_id = 1.location_id)
AND salary > 10000;

C. SELECT emp_name, department_name, city
FROM employees e, departments d, locations 1
WHERE salary > 10000;

D. SELECT emp_name, department_name, city
FROM employees e, departments d, locations 1
WHERE e.department_id = d.department_id
AND d.location_id = 1.location_id
AND salary > 10000;

E. SELECT emp_name, department_name, city
FROM employees e
NATURAL JOIN departments, locations
WHERE salary > 10000;

Q4: Examine the data of the EMPLOYEES table.

EMPLOYEES (EMPLOYEE_ID is the primary key. MGR_ID is the ID of managers and refers to the EMPLOYEE_ID)

EMPLOYEE_ID	EMP_NAME	DEPT_ID	MGR_ID	JOB_ID	SALARY
101	Smith	20	120	SA_REP	4000
102	Martin	10	105	CLERK	2500
103	Chris	20	120	IT_ADMIN	4200
104	John	30	108	HR_CLERK	2500
105	Diana	30	108	HR_MGR	5000
106	Bryan	40	110	AD_ASST	3000
108	Jennifer	30	110	HR_DIR	6500
110	Bob	40		EX_DIR	8000
120	Ravi	20	110	SA_DIR	6500

Which statement lists the ID, name, and salary of the employee, and the ID and name of the employee's manager, for all the employees who have a manager and earn more than 4000?

A. SELECT employee_id "Emp_id", emp_name "Employee",
salary, employee_id "Mgr_id", emp_name "Manager"
FROM employees
WHERE salary > 4000;

B. SELECT e.employee_id "Emp_id", e.emp_name "Employee",
e.salary, m.employee_id "Mgr_id", m.emp_name "Manager"
FROM employees e, employees m
WHERE e.mgr_id = m.mgr_id
AND e.salary > 4000;

C. SELECT e.employee_id "Emp_id", e.emp_name "Employee",
e.salary, m.employee_id "Mgr_id", m.emp_name "Manager"
FROM employees e, employees m
WHERE e.mgr_id = m.employee_id
AND e.salary > 4000;

D. SELECT e.employee_id "Emp_id", e.emp_name "Employee",
e.salary, m.mgr_id "Mgr_id", m.emp_name "manager"
FROM employees e, employees m
WHERE e.mgr_id = m.employee_id
AND e.salary > 4000;

E. SELECT e.employee_id "Emp_id", e.emp_name
"Employee",
e.salary,
m.mgr_id "Mgr_id", m.emp_name "Manager"
FROM employees e, employees m
WHERE e.employee_id = m.employee_id
AND e.salary > 4000;

Q5: Examine the data of the EMPLOYEES table.

EMPLOYEES (EMPLOYEE_ID is the primary key. MGR_ID is the ID of managers and refers to the EMPLOYEE_ID)

EMPLOYEE_ID	EMP_NAME	DEPT_ID	MGR_ID	JOB_ID	SALARY
101	Smith	20	120	SA_REP	4000
102	Martin	10	105	CLERK	2500
103	Chris	20	120	IT_ADMIN	4200
104	John	30	108	HR_CLERK	2500
105	Diana	30	108	HR_MGR	5000
106	Bryan	40	110	AD_ASST	3000
108	Jennifer	30	110	HR_DIR	6500
110	Bob	40		EX_DIR	8000
120	Ravi	20	110	SA_DIR	6500

Evaluate this SQL statement.

```
SELECT e.employee_id "Emp_id", e.emp_name "Employee", e.salary,  
m.employee_id "Mgr_id", m.emp_name "Manager"  
FROM employees e, employees m  
WHERE e.mgr_id = m.employee_id  
AND e.salary > 4000;
```

What is its output?

A.

EMP_id EMPLOYEE SALARY Mgr_id Manager

110 Bob 8000 Bob 120 Ravi 6500 110 Ravi 108 Jennifer 6500 110 Jennifer 103 Chris
4200 120 Chris 105 Diana 5000 108 Diana

B.

EMP_id EMPLOYEE SALARY Mgr_id Manager

120 Ravi 6500 110 Bob 108 Jennifer 6500 110 Bob 103 Chris 4200 120 Ravi 105 Diana
5000 108 Jennifer

C.

EMP_id EMPLOYEE SALARY Mgr_id Manager

110 Bob 8000 120 Ravi 6500 110 Bob 108 Jennifer 6500 110 Bob 103 Chris 4200 120 Ravi 105 Diana 5000 108 Jennifer D EMP_id EMPLOYEE SALARY Mgr_id Manager

110 Bob 8000 110 Bob 120 Ravi 6500 120 Ravi 108 Jennifer 6500 108 Jennifer 103 Chris 4200 103 Chris 105 Diana 5000 105 Dina

E. The SQL statement produces an error.

Q6: In which case would you use a FULL OUTER JOIN?

- A.** Both tables have NULL values.
- B.** You want all unmatched data from one table.
- C.** You want all matched data from both tables.
- D.** You want all unmatched data from both tables.
- E.** One of the tables has more data than the other.
- F.** You want all matched and unmatched data from only one table.

Q7: Examine the structures of the EMPLOYEES and TAX tables. EMPLOYEES

EMPLOYEE_ID	NUMBER	NOT NULL, Primary Key
EMP_NAME	VARCHAR2 (30)	
JOB_ID	VARCHAR2 (20)	
SALARY	NUMBER	
MGR_ID	NUMBER	References EMPLOYEE_ID column
DEPARTMENT_ID	NUMBER	Foreign key to DEPARTMENT_ID column of the DEPARTMENT table

TAX

MIN_SALARY	NUMBER	
MAX_SALARY	NUMBER	
TAX_PERCENT	NUMBER	Percentage tax for given salary range

You need to find the percentage tax applicable for each employee. Which SQL statement **would you use?**

A. SELECT employee_id, salary, tax_percent FROM employees e, tax t WHERE e.salary BETWEEN t.min _ salary AND t.max_salary

B. SELECT employee_id, salary, tax_percent FROM employees e, tax t WHERE e.salary > t.min_salary, tax_percent

C. SELECT employee_id, salary, tax_percent FROM employees e, tax t WHERE MIN(e.salary) = t.min_salary AND MAX(e.salary) = t.max_salary

D. You cannot find the information because there is no common column between the two tables.

Q8: Examine the structure of the EMPLOYEES, DEPARTMENTS, and TAX tables.
EMPLOYEES

EMPLOYEE_ID	NUMBER	NOT NULL, Primary Key
EMP_NAME	VARCHAR2 (30)	
JOB_ID	VARCHAR2 (20)	
SALARY	NUMBER	
MGR_ID	NUMBER	References EMPLOYEE_ID column
DEPARTMENT_ID	NUMBER	Foreign key to DEPARTMENT_ID column of the DEPARTMENTS table

DEPARTMENTS

DEPARTMENT_ID	NUMBER	NOT NULL, Primary Key
DEPARTMENT_NAME	VARCHAR2 30	
MGR_ID	NUMBER	References MGR_ID column of the EMPLOYEES table

TAX

MIN_SALARY	NUMBER
MAX_SALARY	NUMBER
TAX_PERCENT	NUMBER

For which situation would you use a nonequijoin query?

- A. To find the tax percentage for each of the employees.
- B. To list the name, job id, and manager name for all the employees.
- C. To find the name, salary, and department name of employees who are not working with Smith.
- D. To find the number of employees working for the Administrative department and earning less than 4000.
- E. To display name, salary, manager ID, and department name of all the employees, even if the employees do not have a department ID assigned.

Q9: What is true about joining tables through an equijoin?

- A. You can join a maximum of two tables through an equijoin.
- B. You can join a maximum of two columns through an equijoin.
- C. You specify an equijoin condition in the SELECT or FROM clauses of a SELECT statement.
- D. To join two tables through an equijoin, the columns in the join condition must be primary key and foreign key columns.
- E. You can join n tables (all having single column primary keys) in a SQL statement by specifying a minimum of n-1 join conditions.

Q10: Which three are true regarding the use of outer joins? **(Choose three.)**

- A. You cannot use IN operator in a condition that involves an outerjoin.
- B. You use (+) on both sides of the WHERE condition to perform an outerjoin.
- C. You use (*) on both sides of the WHERE condition to perform an outerjoin.
- D. You use an outerjoin to see only the rows that do not meet the join condition.
- E. In the WHERE condition, you use (+) following the name of the column in the table without matching rows, to perform an outerjoin.
- F. You cannot link a condition that is involved in an outerjoin to another condition by using the OR operator.

Q11: Examine the structure of the EMPLOYEES and DEPARTMENTS tables.

EMPLOYEES	EMPLOYEE_ID	NUMBER	DEPARTMENT_ID	NUMBER
MANAGER_ID	NUMBER	LAST_NAME	VARCHAR2(25)	DEPARTMENTS
DEPARTMENT_ID	NUMBER	MANAGER_ID	NUMBER	

DEPARTMENT_NAME VARCHAR2(35) LOCATION_ID NUMBER You want to create a report displaying employee last names, department names, and locations. Which query should you use to create an equi-join?

- A. SELECT last_name, department_name, location_id FROM employees , department ;
- B. SELECT employees.last_name, departments.department_name, departments.location_id FROM employees e, departments D WHERE e.department_id = d.department_id;
- C. SELECT e.last_name, d.department_name, d.location_id FROM employees e, departments D WHERE manager_id = manager_id;
- D. SELECT e.last_name, d.department_name, d.location_id FROM employees e, departments D WHERE e.department_id = d.department_id;

Q12: In which scenario would TOP N analysis be the best solution?

- A. You want to identify the most senior employee in the company.
- B. You want to find the manager supervising the largest number of employees.
- C. You want to identify the person who makes the highest salary for all employees.
- D. You want to rank the top three sales representatives who have sold the maximum number of products.

Q13: In which two cases would you use an outer join? (Choose two.)

- A. The tables being joined have NOT NULL columns.
- B. The tables being joined have only matched data.
- C. The columns being joined have NULL values.
- D. The tables being joined have only unmatched data.
- E. The tables being joined have both matched and unmatched data.
- F. Only when the tables have a primary key/foreign key relationship

EMPLOYEES			
EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	MANAGER_ID
100	Getz	10	103
101	Davis	20	104
102	Bill	20	104
103	Davis	30	
104	Bill		103

DEPARTMENTS	
DEPARTMENT_ID	DEPARTMENT_NAME
10	Sales
20	Marketing
30	Accounts
40	Administration

Q14: Examine the data in the EMPLOYEES and DEPARTMENTS tables. You want to retrieve all employees' last names, along with their manager's last names and their department names. **Which query would you use?**

- A. `SELECT last_name, manager_id, department_name FROM employees e FULL OUTER JOIN department d ON (e.department_id = d.department_id);`
- B. `SELECT e.last_name, m.last_name, department_name FROM employees e LEFT OUTER JOIN employees m on ( e.managaer_id = m.employee_id) LEFT OUTER JOIN department d ON (e.department_id = d.department_id);`
- C. `SELECT e.last_name, m.last_name, department_name FROM employees e RIGHT OUTER JOIN employees m on ( e.manager_id = m.employee_id) FULL OUTER JOIN department d ON (e.department_id = d.department_id);`

D. SELECT e.last_name, m.last_name, department_name FROM employees e LEFT OUTER JOIN employees m on (e.manager_id = m.employee_id) RIGHT OUTER JOIN department d ON (e.department_id = d.department_id);

E. SELECT e.last_name, m.last_name, department_name FROM employees e RIGHT OUTER JOIN employees m on (e.manager_id = m.employee_id) RIGHT OUTER JOIN departments d ON (e.department_id = d.department_id)

F. SELECT last_name, manager_id, department_name FROM employees e JOIN department d ON (e.department_id = d.department_id);

Q15: Evaluate this SQL statement.

```
SELECT e.emp_name, d.dept_name
FROM employees e
JOIN departments d
USING (department_id)
WHERE d.department_id NOT IN (10,40)
ORSER BY dept_name;
```

The statement fails when executed. **Which change fixes the error?**

- A.** remove the ORDER BY clause
 - B.** remove the table alias prefix from the WHERE clause
 - C.** remove the table alias from the SELECT clause
 - D.** prefix the column in the USING clause with the table alias
 - E.** prefix the column in the ORDER BY clause with the table alias
 - F.** replace the condition "d.department_id NOT IN (10,40)" in the WHERE clause with "d.department_id <> 10 AND d.department_id <> 40"
-

With MY Best WiShes

Eng. Sameh Bakkar