Section 3 Programming

CROSS JOINS AND NATURAL JOINS

A NATURAL JOIN is based on:
Columns with the same name and datatype

The ___________ join is the ANSI-standard syntax used to generate a Cartesian product.
CROSS

The join column must be included in the select statement when you use the NATURAL JOIN clause. True or False?
False

What happens when you create a Cartesian product?
All rows from one table are joined to all rows of another table

JOIN CLAUSES

The following is a valid SQL statement.
SELECT e.employee_id, e.last_name, d.location_id, department_id
FROM employees e JOIN departments d
USING (department_id) ;
True or False?
True

The primary advantage of using JOIN ON is:
It permits columns with different names to be joined

Table aliases MUST be used with columns referenced in the JOIN USING clause. True or False?
False

You can do nonequi-joins with ANSI-Syntax. True or False?
True

The keywords JOIN _____________ should be used to join tables with the same column names but different datatypes.
USING

INNER VERSUS OUTER JOINS

What is another name for a simple join or an inner join?

Equijoin

Given the following descriptions of the employees and jobs tables, which of the following scripts will display each employee's possible minimum and maximum salaries based on their job title?

EMPLOYEES Table:

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER (6)
FIRST_NAME		VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (25)
EMAIL	NOT NULL	VARCHAR2 (25)
PHONE_NUMBER		VARCHAR2 (20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2 (10)
SALARY		NUMBER (8,2)
COMMISSION_PCT		NUMBER (2,2)
MANAGER_ID		NUMBER (6)
DEPARTMENT_ID		NUMBER (4)

JOBS Table:

Name	Null?	Type
JOB_ID	NOT NULL	VARCHAR2 (10)
JOB_TITLE	NOT NULL	VARCHAR2 (35)
MIN_SALARY		NUMBER (6)
MAX_SALARY		NUMBER (6)

```
SELECT first_name, last_name, job_id, min_salary, max_salary
FROM employees
NATURAL JOIN jobs;
```

The following statement is an example of what kind of join?

```
SELECT car.vehicle_id, driver.name
FROM car
LEFT OUTER JOIN driver ON (driver_id) ;
```

Outer Join

Which syntax would be used to retrieve all rows in both the EMPLOYEES and DEPARTMENTS tables, even when there is no match?

FULL OUTER JOIN

For which of the following tables will all the values be retrieved even if there is no match in the other?

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

employees

EMPLOYEES Table:

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
DEPARTMENT_ID		NUMBER (4)

DEPARTMENTS Table:

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER 4
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER (6)

A query is needed to display each department and its manager name from the above tables. However, not all departments have a manager but we want departments returned in all cases. Which of the following SQL: 1999 syntax scripts will accomplish the task?

```
SELECT d.department_id, e.first_name, e.last_name
FROM employees e
RIGHT OUTER JOIN departments d
ON (e.employee_id = d.manager_id);
```

If you select rows from two tables (employees and departments) using an outer join, what will you get? Use the code below to arrive at your answer:

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

All employees including those that do not have a departement_id assigned to them

SELF JOINS VERSUS OUTER JOINS

Which of the following database design concepts is implemented with a self join?

Recursive Relationship

Hierarchical queries can walk both Top-Down and Bottom-Up. True or False?

True

Which select statement will return the last name and hire date of an employee and his/ her manager for employees that started in the company before their managers?

```
SELECT w.last_name, w.hire_date, m.last_name, m.hire_date
FROM employees w , employees m
WHERE w.manager_id = m.employee_id
AND w.hire_date < m.hire_date
```

Hierarchical queries MUST use the LEVEL pseudo column. True or False?

False

Which SELECT statement implements a self join?

```
SELECT e.employee_id, m.manager_id
FROM employees e, employees m
WHERE m.employee_id = e.manager_id;
```

Which of the following database design concepts do you need in your tables to write Hierarchical queries?

Recursive Relationship