1. SELECT e.event_name, p.package_code
   FROM d_events e
   JOIN d_packages p ON e.cost = p.cost;
2. SELECT e.last_name, e.salary, j.grade_level
   FROM employees e
   JOIN job_grades j ON e.salary BETWEEN j.lowest_salary AND j.highest_salary;
3. A nonequijoin is required when the join condition compares columns using a range of values or operators other than equality
4. WHERE a.ranking >= g.lowest_rank AND a.ranking <= g.highest_rank;
5. Joining multiple tables, and the tables have columns with the same name, to avoid ambiguity.
6. nonequijoin
7. SELECT c.customer_name, o.order_id
   FROM customers c
   LEFT JOIN orders o ON c.customer_id = o.customer_id;
8. SELECT e.last_name, e.department_id, d.department_name
   FROM employees e
   LEFT JOIN departments d ON e.department_id = d.department_id;
9. SELECT e.last_name, d.department_id, d.department_name
   FROM departments d
   LEFT JOIN employees e ON d.department_id = e.department_id;
10.
    a. WHERE e.department_id(+) = d.department_id;
    b. SELECT e.employee_id, e.last_name, d.location_id
       FROM employees e
       JOIN departments d ON e.department_id = d.department_id;
11. SELECT c.cd_title, t.song_id
    FROM d_cds c
    LEFT JOIN d_track_listings t ON c.cd_id = t.cd_id;
12.