1. **Using `UPDATE` with `DISTINCT` and `ORDER BY`:**

```
-- Update the salary of the highest-paid employee in each department
UPDATE employees e
JOIN (
   SELECT department, MAX(salary) AS max_salary
   FROM employees
   GROUP BY department
) max_salaries
ON e.department = max_salaries.department AND e.salary = max_salaries.max_salary
SET e.salary = e.salary * 1.10;
```

- Explanation: In this example, we're updating the salary of the highest-paid employee in each department. Here's how it works:

   - The subquery selects the maximum salary (`MAX(salary)`) for each department using `GROUP BY`. The result includes the department and the maximum salary (`max_salary`) for each department.

   - We then join this subquery's result with the "employees" table on both the department and salary columns. This allows us to identify the highest-paid employee in each department.

   - Finally, we use `SET` to update the salary of these employees by increasing it by 10% (`e.salary * 1.10`).

2. Using `DISTINCT` and `ORDER BY`:

```
-- Retrieve a list of unique product categories in descending order of popularity (by count)
SELECT DISTINCT category
FROM products
ORDER BY COUNT(*) DESC;
```

- Explanation: In this example, we're retrieving a list of unique product categories in descending order of popularity (by count of products in each category). Here's how it works:

   - We use `DISTINCT` to ensure that each product category appears only once in the result set, eliminating duplicates.

   - We use `ORDER BY` to sort the unique categories by the count of products in each category (`COUNT(*)`) in descending order (`DESC`). This provides a list of categories ordered by popularity.

2. **Combining `UPDATE`, `DISTINCT`, and `ORDER BY`:**

```
-- Update the salaries of the top 5 highest-paid employees across all departments
UPDATE employees e
JOIN (
   SELECT DISTINCT employee_id
   FROM employees
   ORDER BY salary DESC
   LIMIT 5
) top_employees
ON e.employee_id = top_employees.employee_id
SET e.salary = e.salary * 1.05;
```

- Explanation: In this example, we're updating the salaries of the top 5 highest-paid employees across all departments. Here's how it works:

  - The subquery selects distinct `employee_id` values from the "employees" table and orders them by salary in descending order using `ORDER BY`. The `LIMIT 5` clause ensures we only select the top 5 highest-paid employees.

  - We then join this subquery's result with the "employees" table based on the `employee_id`.

  - Finally, we use `SET` to update the salaries of these employees by increasing them by 5% (`e.salary * 1.05`).

These examples demonstrate how you can use `UPDATE`, `DISTINCT`, and `ORDER BY` individually or in combination to perform specific operations on your data, such as updating records based on conditions or retrieving and ordering unique values from a table.