

1. Self-Join – Joins a table to itself.
2. Hierarchical Query – Retrieves data based on a natural hierarchical relationship between rows in a table.
3. LEVEL – Determines the number of steps down from the beginning row that should be returned by a hierarchical query.
4. START WITH – Identifies the beginning row for a hierarchical query.
5. CONNECT BY – Specifies the relationship between parent rows and child rows in a hierarchical query.

1.

```
SELECT e.last_name AS "Employee", e.employee_id AS "Emp#", m.last_name AS
"Manager", m.employee_id AS "Mgr#"
FROM employees e
LEFT JOIN employees m ON e.manager_id = m.employee_id;
```
2.

```
SELECT e.last_name AS "Employee", e.employee_id AS "Emp#", m.last_name AS
"Manager", m.employee_id AS "Mgr#"
FROM employees e
LEFT JOIN employees m ON e.manager_id = m.employee_id
ORDER BY e.last_name;
```
3.

```
SELECT e.last_name AS "Employee", e.hire_date AS "Emp Hired", m.last_name AS
"Manager", m.hire_date AS "Mgr Hired"
FROM employees e
JOIN employees m ON e.manager_id = m.employee_id
WHERE e.hire_date < m.hire_date;
```
4.

```
SELECT last_name, salary, department_id
FROM employees
START WITH last_name = 'De Haan'
CONNECT BY PRIOR employee_id = manager_id;
```
5.

```
SELECT last_name, department_id, salary
FROM employees
START WITH last_name = 'King'
CONNECT BY PRIOR employee_id = manager_id;
```
6.

```
SELECT LPAD('-', 2 * (LEVEL - 1)) || last_name AS "Employee", employee_id,
manager_id
FROM employees
START WITH manager_id IS NULL
CONNECT BY PRIOR employee_id = manager_id;
```
7.

```
SELECT LPAD('-', 2 * (LEVEL - 1)) || last_name AS "Employee", employee_id,
manager_id
FROM employees
START WITH manager_id IS NULL
CONNECT BY PRIOR employee_id = manager_id
AND PRIOR employee_id NOT IN (SELECT employee_id FROM employees WHERE
last_name = 'De Haan');
```