### **1. List all employees who do not report to employee 203 (Martha Woods).**

**SQL Statement:**

SELECT employee\_id, first\_name, last\_name, manager\_id

FROM L\_EMPLOYEES

WHERE manager\_id <> 203;

**Results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **MANAGER\_ID** |
| 202 | JIM | KERN | 201 |
| 203 | MARTHA | WOODS | 201 |
| 204 | ELLEN | OWENS | 202 |
| 205 | HENRY | PERKINS | 202 |
| 209 | PAULA | JACOBS | 201 |

### **2. List all employees who report to employees 202 or 203 (Jim Kern or Martha Woods).**

**SQL Statement:**

SELECT employee\_id, first\_name, last\_name, manager\_id

FROM L\_EMPLOYEES

WHERE manager\_id IN (202, 203);

**Results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **MANAGER\_ID** |
| 204 | ELLEN | OWENS | 202 |
| 205 | HENRY | PERKINS | 202 |
| 207 | DAN | SMITH | 203 |
| 208 | FRED | CAMPBELL | 203 |
| 210 | NANCY | HOFFMAN | 203 |

### **3. Alternative way to write Query 2 using Boolean OR.**

**SQL Statement:**

SELECT employee\_id, first\_name, last\_name, manager\_id

FROM L\_EMPLOYEES

WHERE manager\_id = 202 OR manager\_id = 203;

**Results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **MANAGER\_ID** |
| 204 | ELLEN | OWENS | 202 |
| 205 | HENRY | PERKINS | 202 |
| 207 | DAN | SMITH | 203 |
| 208 | FRED | CAMPBELL | 203 |
| 210 | NANCY | HOFFMAN | 203 |

### **4. List all employees hired between ‘AUG-16-1999’ and ‘JUL-01-2008’.**

**SQL Statement:**

SELECT employee\_id, first\_name, last\_name, hire\_date

FROM L\_EMPLOYEES

WHERE hire\_date BETWEEN TO\_DATE('16-AUG-1999', 'DD-MON-YYYY')

AND TO\_DATE('01-JUL-2008', 'DD-MON-YYYY');

**Results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **HIRE\_DATE** |
| 202 | JIM | KERN | 8/16/1999 |
| 204 | ELLEN | OWENS | 7/1/2008 |
| 205 | HENRY | PERKINS | 3/1/2006 |
| 208 | FRED | CAMPBELL | 4/1/2008 |
| 210 | NANCY | HOFFMAN | 2/16/2007 |

### **5. List employee ID, first name, and last name of employees with IDs between 201 and 205.**

**SQL Statement:**

SELECT employee\_id, first\_name, last\_name

FROM L\_EMPLOYEES

WHERE employee\_id BETWEEN 201 AND 205;

**Results:**

|  |  |  |
| --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** |
| 201 | SUSAN | BROWN |
| 202 | JIM | KERN |
| 203 | MARTHA | WOODS |
| 204 | ELLEN | OWENS |
| 205 | HENRY | PERKINS |

### **6. List all employees who have the letter 'n' in their last name.**

**SQL Statement:**

SELECT employee\_id, first\_name, last\_name

FROM L\_EMPLOYEES

WHERE LOWER(last\_name) LIKE '%n%';

**Results:**

|  |  |  |
| --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** |
| 201 | SUSAN | BROWN |
| 202 | JIM | KERN |
| 204 | ELLEN | OWENS |
| 205 | HENRY | PERKINS |
| 210 | NANCY | HOFFMAN |

### **7. List all employees who have NULL in the manager\_id column.**

**SQL Statement:**

SELECT employee\_id, first\_name, last\_name, manager\_id

FROM L\_EMPLOYEES

WHERE manager\_id IS NULL;

**Results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **MANAGER\_ID** |
| 201 | SUSAN | BROWN | - |
| 206 | CAROL | ROSE | - |

### **8. List last name and first name, renaming last\_name to family\_name, sorted by last name.**

**SQL Statement:**

SELECT last\_name AS family\_name, first\_name

FROM L\_EMPLOYEES

ORDER BY last\_name ASC;

**Results:**

|  |  |
| --- | --- |
| **FAMILY\_NAME** | **FIRST\_NAME** |
| BROWN | SUSAN |
| CAMPBELL | FRED |
| HOFFMAN | NANCY |
| JACOBS | PAULA |
| KERN | JIM |
| OWENS | ELLEN |
| PERKINS | HENRY |
| ROSE | CAROL |
| SMITH | DAN |
| WOODS | MARTHA |

### **9. List department codes and last names of all employees except employee 209, sorted by both columns.**

**SQL Statement:**

SELECT dept\_code, last\_name

FROM L\_EMPLOYEES

WHERE employee\_id <> 209

ORDER BY dept\_code ASC, last\_name ASC;

**Results:**

|  |  |
| --- | --- |
| **DEPT\_CODE** | **LAST\_NAME** |
| ACT | ROSE |
| EXE | BROWN |
| SAL | HOFFMAN |
| SAL | KERN |
| SAL | OWENS |
| SAL | PERKINS |
| SHP | CAMPBELL |
| SHP | SMITH |
| SHP | WOODS |

### **10. List all columns from sec0218 and sort in two different ways.**

**SQL Statement (Method 1 - Sort first by column A, then by column B):**

SELECT A, B

FROM sec0218

ORDER BY A ASC, B ASC;

**Results:**

|  |  |
| --- | --- |
| **A** | **B** |
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 2 | 1 |
| 2 | 2 |
| 2 | 3 |
| 3 | 1 |
| 3 | 2 |
| 3 | 3 |

**SQL Statement (Method 2 - Sort first by column B, then by column A):**

SELECT A, B

FROM sec0218

ORDER BY B ASC, A ASC;

**Results:**

|  |  |
| --- | --- |
| **A** | **B** |
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 1 | 2 |
| 2 | 2 |
| 3 | 2 |
| 1 | 3 |
| 2 | 3 |
| 3 | 3 |