#### **(11-3) Question 1: Find the following:**

* The minimum credit limit given to any employee
* The maximum credit limit given to any employee
* The first name of an employee that comes last alphabetically
* The last name of an employee that comes last alphabetically
* The latest date when any of the employees was hired

**Query:**  
SELECT  
MIN(credit\_limit) AS min\_credit\_limit,  
MAX(credit\_limit) AS max\_credit\_limit,  
MAX(first\_name) AS last\_first\_name,  
MAX(last\_name) AS last\_last\_name,  
MAX(hire\_date) AS latest\_hire\_date  
FROM L\_EMPLOYEES;  
-- Haley Archer

**Output:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MIN\_CREDIT\_LIMIT** | **MAX\_CREDIT\_LIMIT** | **LAST\_FIRST\_NAME** | **LAST\_LAST\_NAME** | **LATEST\_HIRE\_DATE** |
| 15 | 30 | SUSAN | WOODS | 2/2/2009 |

#### **(11-5) Question 2: Find all employees who have the minimum credit limit.**

**Query:**  
SELECT employee\_id, first\_name, last\_name, credit\_limit  
FROM L\_EMPLOYEES  
WHERE credit\_limit = (SELECT MIN(credit\_limit) FROM L\_EMPLOYEES);  
-- Haley Archer

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **CREDIT\_LIMIT** |
| 204 | ELLEN | OWENS | 15 |
| 209 | PAULA | JACOBS | 15 |

#### **(11-7) Question 3: Apply all column functions to a column that contains only nulls.**

**Query:**  
SELECT  
COUNT(NULL\_COLUMN) AS count\_result,  
MIN(NULL\_COLUMN) AS min\_result,  
MAX(NULL\_COLUMN) AS max\_result,  
SUM(NULL\_COLUMN) AS sum\_result,  
AVG(NULL\_COLUMN) AS avg\_result  
FROM (SELECT NULL AS NULL\_COLUMN FROM DUAL CONNECT BY LEVEL <= 10);  
-- Haley Archer

**Output:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COUNT\_RESULT** | **MIN\_RESULT** | **MAX\_RESULT** | **SUM\_RESULT** | **AVG\_RESULT** |
| 0 | - | - | - | - |

#### **(11-9) Question 4: Count the number of distinct combinations of manager\_id and credit\_limit.**

**Query:**  
SELECT COUNT(DISTINCT manager\_id || '-' || credit\_limit) AS distinct\_combinations  
FROM L\_EMPLOYEES;  
-- Haley Archer

**Output:**

|  |
| --- |
| **DISTINCT\_COMBINATIONS** |
| 7 |

#### **(11-12) Question 5: Find the total for all the invoices in SEC1112\_SHIPPING table.**

*Estimate values for nulls in tax and shipping using:*

1. Replace a null in the tax column with 0.07 \* price \* quantity
2. Replace a null in the shipping column with 0.12 \* price \* quantity

**Query:**  
SELECT PK\_1 AS invoice\_id,  
(price \* quantity) +  
COALESCE(tax, 0.07 \* price \* quantity) +  
COALESCE(shipping, 0.12 \* price \* quantity) AS total\_invoice  
FROM SEC1112\_SHIPPING;  
-- Haley Archer

**Output:**

|  |  |
| --- | --- |
| **INVOICE\_ID** | **TOTAL\_INVOICE** |
| A | 744 |
| B | 1105.62 |
| C | 719 |
| D | 790.96 |

#### **(11-14) Question 6: Count the number of different dates in the date\_entered column of L\_LUNCHES table.**

**Query:**  
SELECT COUNT(DISTINCT date\_entered) AS unique\_dates  
FROM L\_LUNCHES;  
-- Haley Archer

**Output:**

|  |
| --- |
| **UNIQUE\_DATES** |
| 16 |

#### **(12-2) Question 7: For each manager\_id, list the number of employees each one manages.**

* Also list the range of their employees’ credit limits (min/max).
* Omit employee 202.

**Query:**  
SELECT manager\_id,  
COUNT(\*) AS num\_employees,  
MIN(credit\_limit) AS min\_credit\_limit,  
MAX(credit\_limit) AS max\_credit\_limit  
FROM L\_EMPLOYEES  
WHERE employee\_id <> 202  
GROUP BY manager\_id;  
-- Haley Archer

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **MANAGER\_ID** | **NUM\_EMPLOYEES** | **MIN\_CREDIT\_LIMIT** | **MAX\_CREDIT\_LIMIT** |
| 203 | 3 | 25 | 25 |
| - | 2 | 30 | 30 |
| 202 | 2 | 15 | 25 |
| 201 | 2 | 15 | 25 |

#### **(12-5) Question 8: Show error when mixing a summarized column with a non-summarized column.**

**Query:**  
SELECT manager\_id,  
COUNT(\*) AS num\_employees,  
MAX(credit\_limit) AS max\_credit\_limit  
FROM L\_EMPLOYEES  
GROUP BY manager\_id;  
-- Haley Archer

**Output:**

|  |  |  |
| --- | --- | --- |
| **MANAGER\_ID** | **NUM\_EMPLOYEES** | **MAX\_CREDIT\_LIMIT** |
| 203 | 3 | 25 |
| - | 2 | 30 |
| 202 | 2 | 25 |
| 201 | 3 | 25 |

#### **(12-9) Question 9: Count the number of lunches each employee will attend.**

**Query:**  
SELECT employee\_id, COUNT(\*) AS num\_lunches  
FROM L\_LUNCHES  
GROUP BY employee\_id;  
-- Haley Archer

**Output:**

|  |  |
| --- | --- |
| **EMPLOYEE\_ID** | **NUM\_LUNCHES** |
| 210 | 2 |
| 203 | 2 |
| 204 | 2 |
| 202 | 1 |
| 201 | 3 |
| 207 | 2 |
| 208 | 2 |
| 205 | 2 |

#### **(12-10) Question 10: Count the number of lunches each employee will attend, listing employee IDs and names.**

**Query:**  
SELECT e.employee\_id, e.first\_name, e.last\_name, COUNT(l.lunch\_id) AS num\_lunches  
FROM L\_EMPLOYEES e  
LEFT JOIN L\_LUNCHES l ON e.employee\_id = l.employee\_id  
GROUP BY e.employee\_id, e.first\_name, e.last\_name;  
-- Haley Archer

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **NUM\_LUNCHES** |
| 201 | SUSAN | BROWN | 3 |
| 202 | JIM | KERN | 1 |
| 203 | MARTHA | WOODS | 2 |
| 204 | ELLEN | OWENS | 2 |
| 205 | HENRY | PERKINS | 2 |
| 206 | CAROL | ROSE | 0 |
| 207 | DAN | SMITH | 2 |
| 208 | FRED | CAMPBELL | 2 |
| 209 | PAULA | JACOBS | 0 |
| 210 | NANCY | HOFFMAN | 2 |