* **Question 1**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | * 1. Display the names of the employees whose salary is the same as the lowest salaried employee **in any department**. |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | SELECT first\_name || ' ' || last\_name AS Employee  FROM   employees  WHERE  salary = (                  SELECT MIN(salary)                  FROM   employees                  )  ORDER BY Employee;  EMPLOYEE  ----------------------------------------------  Jennifer Whalen  Peter Vargas  Randall Matos | | Correct Answer: | Correct  SELECT (first\_name || ' ' || last\_name) "Name"  FROM employees  WHERE salary = any (    SELECT MIN(salary)    FROM employees    GROUP BY department\_id    )  AND department\_id IS NOT NULL;  The intent is to use ANY as that word appears in the question, but either is correct  SELECT (first\_name || ' ' || last\_name) "Name"  FROM employees  WHERE salary IN (    SELECT MIN(salary)    FROM employees    GROUP BY department\_id    )  AND department\_id IS NOT NULL;    Using IN | |  |  |  |

* **Question 2**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Create a view named ALLEMPS that consists of all employees includes employee\_id, last\_name, salary, department\_id, department\_name, city and country (if applicable) |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | CREATE VIEW AllEMPS AS  SELECT e.employee\_id, e.last\_name, e.salary,         department\_id,         d.department\_name,         l.city,         c.country\_name  FROM   employees e  JOIN   departments d  USING (department\_id)  JOIN   locations   l  USING (location\_id)  JOIN   countries   c  USING (country\_id); | | Correct Answer: | Correct  **CREATE VIEW ALLEMPS as**  **SELECT e.employee\_id, e.last\_name, e.salary, e.department\_id,**  **d.department\_name, l.city, l.country\_id**  **From employees e**  **LEFT JOIN departments d**  **ON e.department\_id = d.department\_id**  **LEFT JOIN**  **locations l**  **ON d.location\_id = l.location\_id**  **To see the result enter**  **SELECT \* FROM ALLEAMPS;  A BIT TOO WIDE TO DISPLAY TITLES NICELY** | |  |  |  |

* **Question 3**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Use the ALLEMPS view to:   * 1. Display the employee\_id, last\_name, salary and city for all employees |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | SELECT   employee\_id, last\_name, salary, city  FROM     allemps  ORDER BY employee\_id;  EMPLOYEE\_ID LAST\_NAME                     SALARY CITY  ----------- ------------------------- ---------- ------------------------------            1 Flertjan                       11000 Oxford            3 Grovlin                        11000 Oxford            4 Smertal                        11000 Oxford            5 Mustaine                       11000 Oxford            6 Harvey                         10000 Oxford            7 LeDuc                           7000 Oxford            8 Bergsteige                      8000 Oxford            9 Gruber                          9000 Oxford           11 Sanchez                        11000 Oxford           12 Chancevente                    12000 Oxford           14 Torson                         11000 Oxford                          EMPLOYEE\_ID LAST\_NAME                     SALARY CITY  ----------- ------------------------- ---------- ------------------------------           15 Cornel                         11000 Oxford           16 Gibbons                        11000 Oxford           17 Pallomine                      11000 Oxford           18 Jacobs                         11000 Oxford           19 Strandherst                     9000 Oxford           21 Brigade                        11000 Oxford           22 Litrand                        10000 Oxford           23 Armarillo                      11000 Oxford           24 Mot                            11000 Oxford           25 Turcotte                       11000 Oxford           26 LeBlanc                        11000 Oxford                          EMPLOYEE\_ID LAST\_NAME                     SALARY CITY  ----------- ------------------------- ---------- ------------------------------           27 Rodriguez                      11000 Oxford           28 Young                          10000 Oxford           29 Loo Nam                        11000 Oxford           30 Chan                           11000 Oxford           33 Wandiko                        11000 Oxford           34 Gregson                        11000 Oxford           35 Krain                          11000 Oxford           36 Termede                        11000 Oxford           39 Testorok                       11000 Oxford           40 Whiteduck                      11000 Oxford           41 Montoya                        11000 Oxford                          EMPLOYEE\_ID LAST\_NAME                     SALARY CITY  ----------- ------------------------- ---------- ------------------------------          100 King                           24000 Seattle          101 Kochhar                        17000 Seattle          102 De Haan                        17000 Seattle          103 Hunold                          9000 Southlake          104 Ernst                           6000 Southlake          107 Lorentz                         4200 Southlake          124 Mourgos                         5800 South San Francisco          141 Rajs                            3500 South San Francisco          142 Davies                          3100 South San Francisco          143 Matos                           2500 South San Francisco          144 Vargas                          2500 South San Francisco             EMPLOYEE\_ID LAST\_NAME                     SALARY CITY  ----------- ------------------------- ---------- ------------------------------          149 Zlotkey                        10500 Oxford          174 Abel                           11000 Oxford          176 Taylor                          8600 Oxford          200 Whalen                          2500 Seattle         201 Hartstein                      13000 Toronto         202 Fay                             6000 Toronto         205 Higgins                        12000 Seattle         206 Gietz                           8300 Seattle         207 Akhlaqi                              Seattle  53 rows selected. | | Correct Answer: | Correct  **SELECT employee\_id, last\_name, salary, city**  **FROM allemps**  **ORDER BY employee\_id;** | |  |  |  |

* **Question 4**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Using the same view  Display the total salary of all employees by city |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | SELECT city, SUM(salary)  FROM allemps  GROUP BY city  ORDER BY city;  CITY                           SUM(SALARY)  ------------------------------ -----------  Oxford                              380100  Seattle                              80800  South San Francisco                  17400  Southlake                            19200  Toronto                              19000 | | Correct Answer: | Correct  SELECT city AS "City", SUM(salary) as "Salary"  FROM allemps  GROUP BY city; | |  |  |  |

* **Question 5**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Increase the salary of the lowest paid employee(s) in each department by 100 |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | UPDATE allemps  SET    salary = salary + 100  WHERE  employee\_id IN (                         SELECT employee\_id                         FROM   employees                         WHERE  (department\_id, salary) IN (                                                            SELECT   department\_id, MIN(salary)                                                            FROM     employees                                                            GROUP BY department\_id                                                            )                         );  9 rows updated. | | Correct Answer: | Correct  **UPDATE employees**  **SET SALARY = SALARY + 100** **WHERE salary IN (** **SELECT MIN(salary)** **FROM allemps** **WHERE department\_id IS NOT NULL** **GROUP BY department\_id);**    **Proof -----**  **SELECT first\_name||' '||last\_name "Name", salary**  **FROM employees**  **WHERE (salary, department\_id) IN (**  **SELECT MIN(salary), department\_id**  **FROM employees**  **WHERE department\_id IS NOT NULL**  **GROUP BY department\_id);** | |  |  |  |

* **Question 6**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | What happens if you try to insert an employee by providing values for all columns in this view? |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | INSERT INTO allemps VALUES(300, 'mansour', 5000, 10, 'Administration', 'Toronto', 'Canada');  it's not possible to modify more than one base table through a join view. | | Correct Answer: | Correct  **Error starting at line : 1 in command -**  **INSERT INTO allemps**  **VALUES (999,'Voigt',15000,60,'IT','Southlake','US')**  **Error at Command Line : 1 Column : 1**  **Error report -**  **SQL Error: ORA-01776: cannot modify more than one base table through a join view**  **01776. 00000 -  "cannot modify more than one base table through a join view"**  **\*Cause:    Columns belonging to more than one underlying table were either**  **inserted into or updated.**  **\*Action:   Phrase the statement as two or more separate statements.** | |  |  |  |

* **Question 7**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | DELETE the employee named Vargas |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | DELETE FROM ALLEMPS  WHERE last\_name = 'Vargas';  1 row deleted. | | Correct Answer: | Correct  **DELETE FROM allemps**  **WHERE last\_name = 'Vargas';**    **Running SELECT \* FROM ALLEMPS will show 19 employees** | |  |  |  |

* **Question 8**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Create a view named ALLDEPTS that consists of all departments and includes department\_id, department\_name, city and country (if applicable) |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | CREATE VIEW ALLDEPTS AS  SELECT department\_id, department\_name, city, country\_name  FROM   departments  JOIN locations USING (location\_id)  JOIN countries USING (country\_id);  View ALLDEPTS created. | | Correct Answer: | Correct  CREATE VIEW ALLDEPTS AS    SELECT department\_id, department\_name, city, country\_id    FROM departments    LEFT JOIN locations USING (location\_id)    LEFT JOIN countries USING (country\_id)    ORDER BY department\_id; | |  |  |  |

* **Question 9**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | For all departments display the department\_id, name and city. Use ALLDEPTS |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | SELECT department\_id,         department\_name,         city  FROM   alldepts  ORDER BY department\_id;  DEPARTMENT\_ID DEPARTMENT\_NAME                CITY  ------------- ------------------------------ ------------------------------             10 Administration                 Seattle             20 Marketing                      Toronto             50 Shipping                       South San Francisco             60 IT                             Southlake             80 Sales                          Oxford             90 Executive                      Seattle            110 Accounting                     Seattle            190 Contracting                    Seattle  8 rows selected. | | Correct Answer: | Correct  **SELECT department\_id, department\_name, city**  **FROM alldepts;**    **DEPARTMENT\_ID DEPARTMENT\_NAME                CITY**  **------------- ------------------------------ ------------------------------**  **60 IT                             Southlake**  **50 Shipping                       South San Francisco**  **190 Contracting                    Seattle**  **110 Accounting                     Seattle**  **90 Executive                      Seattle**  **10 Administration                 Seattle**  **20 Marketing                      Toronto**  **80 Sales                          Oxford** | |  |  |  |

* **Question 10**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | For each city that has departments located in it display the number of departments |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | SELECT city, COUNT(department\_id)  FROM alldepts  GROUP BY city  ORDER BY city;  CITY                           COUNT(DEPARTMENT\_ID)  ------------------------------ --------------------  Oxford                                            1  Seattle                                           4  South San Francisco                               1  Southlake                                         1  Toronto                                           1 | | Correct Answer: | Correct  SELECT city, COUNT(department\_id)  FROM ALLDEPTS  GROUP BY city; | |  |  |  |

* **Question 11**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Create a view called ALLDEPTSUMM that consists of all departments and includes for each department: department\_id, department\_name, number of employees, number of salaried employees, total salary of all employees |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | CREATE VIEW ALLDEPTSUMM AS  SELECT d.department\_id,         d.department\_name,         COUNT(e.employee\_id) AS "EMPS\_NUM",         COUNT(c.employee\_id) AS "EMPS\_SALARIED\_NUM",         SUM(e.salary) AS "EMPS\_TOTAL\_SALARY"  FROM departments d  LEFT JOIN employees e  ON d.department\_id = e.department\_id  LEFT JOIN employees c  ON e.employee\_id = c.employee\_id  GROUP BY d.department\_id, d.department\_name;    View ALLDEPTSUMM created. | | Correct Answer: | Correct  Important question to ask.  Are the “number of employees” the same as “number of salaried employees”?  All of the employees have salaries, so shouldn’t they all be “salaried employees”?  The SQL below assumed they were the same. BUT, presumably they must be different as why do the same thing twice. The difference is some have commission and therefore are not salaried.    CREATE VIEW ALLDEPTSUMM AS    SELECT department\_id, department\_name, COUNT(employee\_id) "NO\_OF\_EMPS",    COUNT(employee\_id) "NO\_OF\_SALARIED\_EMPS",    SUM(salary) "TOTAL\_SALARY"    FROM departments JOIN employees USING (department\_id)    GROUP BY department\_id, department\_name    ORDER BY department\_id; | |  |  |  |

* **Question 12**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Use the ALLDEPTSUMM view to display department name and number of employees for departments that have more than the average number of employees |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | SELECT department\_name, EMPS\_NUMFROM alldeptsumm  WHERE EMPS\_NUM > (                    SELECT AVG(EMPS\_NUM)                    FROM alldeptsumm                    )  ORDER BY department\_name;  DEPARTMENT\_NAME                  EMPS\_NUM  ------------------------------ ----------  Sales                                  36 | | Correct Answer: | Correct  **SELECT department\_name, no\_of\_emps**  **FROM ALLDEPTSUMM**  **WHERE no\_of\_emps > (SELECT AVG(no\_of\_emps)**  **FROM ALLDEPTSUMM);** | |  |  |  |

* **Question 13**

Needs Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | In the past this has not worked, so this for reading only   * 1. GRANT statement to allow another student (use their Neptune account) to   2. retrieve data for your employees table and   3. to allow that same student to retrieve, insert and update data in your departments table     **ORA-01031: insufficient privileges**  **GRANT SELECT ON allemps TO belvedere;  belvedere was used as an example**    **GRANT SELECT, UPDATE, INSERT ON alldepts TO belvedere;**  **11 Use the REVOKE statement to remove permission for me to insert and update data in your departments table**    **ORA-01031: insufficient privileges;**  **REVOKE INSERT, UPDATE ON alldepts FROM belvedere;** |  |  |  |
| |  |  | | --- | --- | | Selected Answer: | noted it. | | Correct Answer: | [None] | |  |  |  |