****

**UPA**

**Universidad Politécnica de Aguascalientes.**

**ISC06B**

**DATA BASE ADMINISTRATION**

# DP 20 PRACTICE

**TEACHER: Juan Carlos Herrera Hernández**

**STUDENTS:**

**Uriel Isaac Vazquez Martínez UP210934**

**Juan Pablo López González UP200053**

**José Miguel Escalera Rubalcava  UP200667**

**Derek Gilberto Ramírez López UP200424**

**Aguascalientes, Ags, November 12th, 2022**

**Database Programming with SQL**

**20-1: Ensuring Quality Query Results – Advanced Techniques**

**1. Create the additional tables used in this reduced section with the following statements:**

|  |  |
| --- | --- |
| Problem No: 1 | No. Rows in Result: |
| Create the additional tables used in this reduced section with the following statements: | 0 |
| Text Code (No image) : | |
| CREATE TABLE emp AS SELECT \* FROM employees;  CREATE TABLE dept AS SELECT \* FROM departments; | |
| Image Result: | |
|  | |

**2. Create a report that displays the constraint name, type, column name, and column position of all constraints in the JOB\_HISTORY table, in addition to those nonzero constraints.**

|  |  |
| --- | --- |
| Problem No: 1 | No. Rows in Result: |
| Create a report that displays the constraint name, type, column name, and column position of all constraints in the JOB\_HISTORY table, in addition to those not null constraints. | 5 |
| Text Code (No image) : | |
| select a.CONSTRAINT\_NAME, a.CONSTRAINT\_TYPE, b.column\_name, b.position  from user\_constraints a inner join user\_cons\_columns b on (a.constraint\_name = b.constraint\_name)  where a.table\_name = 'JOB\_HISTORY'  and a.CONSTRAINT\_TYPE <> 'C'; | |
| Image Result: | |
|  | |

**3. Create a primary key constraint on the employee\_id column of the emp table**.

|  |  |
| --- | --- |
| Problem No: 3 | No. Rows in Result: |
| Create a primary key constraint on the employee\_id column of the emp table. | 0 |
| Text Code (No image) : | |
| ALTER TABLE emp ADD CONSTRAINT "EMP\_PRIMARY\_KEY" PRIMARY KEY (EMPLOYEE\_ID); | |
| Image Result: | |
|  | |

**4. Create a primary key on the department\_id column of the dept table.**

|  |  |
| --- | --- |
| Problem No: 4 | No. Rows in Result: |
| Create a primary key constraint on the employee\_id column of the emp table. | 0 |
| Text Code (No image) : | |
| ALTER TABLE dept ADD CONSTRAINT "DEPT\_PRIMARY\_KEY" PRIMARY KEY (DEPARTMENT\_ID); | |
| Image Result: | |
|  | |

**5. Add a foreign constraint between DEPT and EMP, so that onlyvalid departments in the EMP table. Make sure you can delete any row in the DEPT table and that the referenced rows in the EMP table are deleted.**

|  |  |
| --- | --- |
| Problem No: 5 | No. Rows in Result: |
| Add a foreign constraint between DEPT and EMP, so that onlyvalid departments in the EMP table. Make sure you can delete any row in the DEPT table and that the referenced rows in the EMP table are deleted. | 0 |
| Text Code (No image) : | |
| ALTER TABLE emp ADD CONSTRAINT "EMP\_DEPT\_ID\_FK" FOREIGN KEY (department\_id)  REFERENCES dept(department\_id)  on delete cascade; | |
| Image Result: | |
|  | |

**6. Test the foreign key constraint you just created: Count the number of rows in the EMP table. Remove department 10 from the dept table. Now count emps again. There should be fewer employees.**

|  |  |
| --- | --- |
| Problem No: 6.1 | No. Rows in Result: |
| Count the number of rows in the EMP table | 1 |
| Text Code (No image) : | |
| select count(\*) as "Num emps"  from emp; | |
| Image Result: | |
|  | |

|  |  |
| --- | --- |
| Problem No: 6.2 | No. Rows in Result: |
| Remove department 10 from the dept table. | 0 |
| Text Code (No image) : | |
| delete dept  where department\_id=10; | |
| Image Result: | |
|  | |

|  |  |
| --- | --- |
| Problem No: 6.3 | No. Rows in Result: |
| Now count emps again | 1 |
| Text Code (No image) : | |
| select count(\*) as "Num emps"  from emp; | |
| Image Result: | |
|  | |

**7. Produce a report that returns the last name, salary, department number, and average salary of all the departments where salary is greater than the average salary.**

|  |  |
| --- | --- |
| Problem No: 7 | No. Rows in Result: |
| Produce a report that returns the last name, salary, department number, and average salary of all the departments where salary is greater than the average salary. | 7 |
| Text Code (No image) : | |
| select last\_name,salary, e.department\_id, tab.avgsalary  from employees e , (select department\_id, floor(avg(salary)) avgsalary from employees  group by department\_id) tab  where e.salary > tab.avgsalary and e.department\_id = tab.department\_id; | |
| Image Result: | |
|  | |

**8. Create a view named V2 that returns the highest salary, lowest salary, average salary and department name. Tables used: -emp, dept**

|  |  |
| --- | --- |
| Problem No: 8 | No. Rows in Result: |
| Create a view named V2 that returns the highest salary, lowest salary, average salary and department name | 6 |
| Text Code (No image) : | |
| CREATE OR REPLACE VIEW V2 AS  select d.department\_name "Nombre del Departamento",  min(salary) "Lowest Salary", max(salary) as "Max Salary",  floor(avg(e.salary)) "Average Salary" from emp e  inner join dept d on e.department\_id = d.department\_id  where e.department\_id is not null  group by d.department\_name;  select \* from v2; | |
| Image Result: | |
|  | |

**9. Create a view named Dept\_Managers\_view that returns a listing of department names long with the manager initial and surname for that department. Test the view by returning all the rows from it. Make sure no rows can be updated through the view. Try to run an UPDATE statement against the view.**

|  |  |
| --- | --- |
| Problem No: 9 | No. Rows in Result: |
| Create a view named Dept\_Managers\_view that returns a listing of department names long with the manager initial and surname for that department. Test the view by returning all the rows from it. Make sure no rows can be updated through the view. Try to run an UPDATE statement against the view. | 7 |
| Text Code (No image) : | |
| CREATE OR REPLACE VIEW Dept\_Managers\_view AS  Select d.department\_name, (substr(e.first\_name,0,1) || '.' ||e.last\_name) as MGR\_NAME  FROM employees e  inner join departments d on e.department\_id = d.department\_id  where d.manager\_id = e.employee\_id  WITH READ ONLY;  select \* from DEPT\_MANAGERS\_VIEW;  UPDATE DEPT\_MANAGERS\_VIEW  SET department\_name = 'Exec'  WHERE department\_name= 'Executive'; | |
| Image Result: | |
|  | |

**10. −The following statement contains errors: DROP V3 views;**

**−Fix them and run the code to get the displayed result**

|  |  |
| --- | --- |
| Problem No: 10 | No. Rows in Result: |
| Correct the following code to drop the view V3 | 0 |
| Text Code (No image) : | |
| drop view V3; | |
| Image Result: | |
|  | |

**11. Create a sequence named ct\_seq using all the default values. Run the statements and fix the error. Correct the statement to return the subsequent number**

|  |  |
| --- | --- |
| Problem No: 11 | No. Rows in Result: |
| Create a sequence named ct\_seq using all the default values. Correct the sequence to return the subsequent number | 1 |
| Text Code (No image) : | |
| CREATE SEQUENCE ct\_seq;  select ct\_seq.NEXTVAL from dual;  select ct\_seq.CURRVAL from dual; | |
| Image Result: | |
|  | |

**12. Examine the following insert statement and fix the errors.**

**INSERT INTO emp**

**(employee\_id, first\_name, last\_name, email, phone\_number, hire\_date,**

**job\_id, salary, commission\_pct, manager\_id, department\_id)**

**VALUES**

**(ct\_seq.nextvalue, "Kaare", 'Hansen', 'KHANSEN', '44965 832123',**

**sysdate, 'SA\_REP', $6500, null, 100, 20);**

|  |  |
| --- | --- |
| Problem No: 12 | No. Rows in Result: |
| Examine the following insert statement and fix the errors. | 0 |
| Text Code (No image) : | |
| Insert into emp  (employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id, salary,commission\_pct,manager\_id,department\_id)  Values  (ct\_seq.currval, 'Kaare','Hansen','KHANSEN','4496583212', sysdate, 'SA\_REP', 6500, null,100,20); | |
| Image Result: | |
|  | |

**13. Fix the error in the SQL statement to create the index as shown in the screenshot:**

**CREATE INX emp indx FOR TABLE emp(employee\_id DESC,**

**UPPR(SUBST(firstname,1.1 ||" "||astname)**

|  |  |
| --- | --- |
| Problem No: 13 | No. Rows in Result: |
| Fix the error in the SQL statement to create the index as shown | 0 |
| Text Code (No image) : | |
| CREATE INDEX emp\_indx ON emp(employee\_id DESC,  UPPER(SUBSTR(first\_name,1,1) ||' '||last\_name)); | |
| Image Result: | |
|  | |

|  |  |
| --- | --- |
| Problem No: 13.1 | No. Rows in Result: |
| See index | 2 |
| Text Code (No image) : | |
| select a.table\_name, index\_name,b.index\_type, a.column\_position  from all\_ind\_columns a inner join user\_indexes b using (index\_name)  where index\_name like 'EMP\_INDX'; | |
| Image Result: | |
|  | |

**14. Write the SQL statement to list all the user tables which contain the name PRIV**

|  |  |
| --- | --- |
| Problem No: 14 | No. Rows in Result: |
| Write the SQL statement to list all the user tables which contain the name PRIV | 33 |
| Text Code (No image) : | |
| select table\_name, comments  from dictionary  where table\_name like '%PRIV%'; | |
| Image Result: (next\_page) | |
|  | |

**15. Give select access to public on the EMP table, and verify the grant by running this query.**

|  |  |
| --- | --- |
| Problem No: 15 | No. Rows in Result: |
| Give select access to public on the EMP table, and verify the grant by running this query | 0 |
| Text Code (No image) : | |
| grant select on emp to public;  SELECT \* FROM USER\_TAB\_PRIVS  where table\_name like 'EMP'; | |
| Image Result: | |
|  | |

**16. Replace the ?? in the following query using regular expressions to return only the numbers from the following string: 'Oracle Academy9547d6905%&^ db apex'.**

**SELECT REGEXP\_REPLACE('Oracle Academy9547d6905%&^ db apex',??,'') regexpreplace**

**FROM DUAL;**

|  |  |
| --- | --- |
| Problem No: 16 | No. Rows in Result: |
| Replace the ?? in the following query using regular expressions to return only the numbers from the following string: 'Oracle Academy9547d6905%&^ db apex'. | 1 |
| Text Code (No image) : | |
| SELECT REGEXP\_REPLACE('Oracle Academy9547d6905%&^ db apex','\D','') regexpreplace  FROM DUAL; | |
| Image Result: | |
|  | |

**17. Amend the previous query using regular expressions to return the number of digits from the following string: 'Oracle Academy9547d6905%&^ db’**

**SELECT LENGTH(REGEXP\_REPLACE('Oracle Academy9547d6905%&^ db apex','??','')) regexpreplace**

**FROM DUAL;**

|  |  |
| --- | --- |
| Problem No: 17 | No. Rows in Result: |
| Amend the previous query using regular expressions to return the number of digits from the following string: 'Oracle Academy9547d6905%&^ db’ | 1 |
| Text Code (No image) : | |
| SELECT LENGTH(REGEXP\_REPLACE('Oracle Academy9547d6905%&^ db apex','\D','')) regexpreplace  FROM DUAL; | |
| Image Result: | |
|  | |

**18. Amend the query again to return only the non-numeric characters.**

**SELECT REGEXP\_REPLACE('Oracle Academy9547d6905%&^ db apex','??','') regexpreplace**

**FROM DUAL;**

|  |  |
| --- | --- |
| Problem No: 18 | No. Rows in Result: |
| **Amend the query again to return only the non-numeric characters.** | 1 |
| Text Code (No image) : | |
| SELECT REGEXP\_REPLACE('Oracle Academy9547d6905%&^ db apex','\d','') regexpreplace  FROM DUAL; | |
| Image Result: | |
|  | |

**19. Using Oracle proprietary joins, construct a statement that returns all the employee\_ids joined to**

**all the department\_names.**

|  |  |
| --- | --- |
| Problem No: 19 | No. Rows in Result: |
| Using Oracle proprietary joins, construct a statement that returns all the employee\_ids joined to all the department\_names. | 19 |
| Text Code (No image) : | |
| select employee\_id, department\_name  from employees cross join departments; | |
| Image Result: | |
|  | |

**20. Still using Oracle Joins, correct the previous statement so that it returns only the name of the**

**department that the employee actually works in.**

|  |  |
| --- | --- |
| Problem No: 20 | No. Rows in Result: |
| Still using Oracle Joins, correct the previous statement so that it returns only the name of the department that the employee actually works in. | 19 |
| Text Code (No image) : | |
| select e.employee\_id , d.department\_name  from employees e, departments d  where e.department\_id= d.department\_id  and d.department\_id is not null; | |
| Image Result: | |
|  | |

**21. Still using Oracle Joins, construct a query that lists the employees last name, the department name, the salary, and the country name of all employees.**

|  |  |
| --- | --- |
| Problem No: 21 | No. Rows in Result: |
| Still using Oracle Joins, construct a query that lists the employees last name, the department name, the salary, and the country name of all employees. | 19 |
| Text Code (No image) : | |
| select e.last\_name, d.department\_name, e.salary, c.country\_name  from employees e inner join departments d using (department\_id)  inner join locations l using (location\_id)  inner join countries c using (country\_id); | |
| Image Result: | |
|  | |

**22. Still using Oracle join syntax, alter the previous query so that it also includes the employee record of the employee with no department\_id, ‘Grant’.**

|  |  |
| --- | --- |
| Problem No: 22 | No. Rows in Result: |
| Still using Oracle join syntax, alter the previous query so that it also includes the employee record of the employee with no department\_id, ‘Grant’. | 20 |
| Text Code (No image) : | |
| select e.last\_name, d.department\_name, e.salary, c.country\_name  from employees e left outer join departments d using (department\_id)  left outer join locations l using (location\_id)  left outer join countries c using (country\_id); | |
| Image Result: | |
|  | |