

Práctica 05

Sobre uso de SQL Avanzado para obtención de estadísticas e implementación de restricciones de integridad

PLANTILLA DE RESPUESTAS

En cada item colocar el código del comando SQL implementado.

ITEM	RESPUESTA
PARTE 1: Reportes estadísticos sobre esquema HR	
1)	<pre>-- 1 Cree una consulta para mostrar el número total de empleados y, -- de ese total, el número de empleados contratados en cada año. Cree encabezados de columna apropiados. SELECT YEAR, EMPLEADOS_POR_AÑO AS "Empleados_Contratados_Por_Año" FROM (SELECT TO_CHAR(HIRE_DATE, 'YYYY') AS YEAR, COUNT(*) AS EMPLEADOS_POR_AÑO FROM EMPLOYEES GROUP BY TO_CHAR(HIRE_DATE, 'YYYY') ORDER BY TO_CHAR(HIRE_DATE, 'YYYY')) GROUP BY YEAR, EMPLEADOS_POR_AÑO ORDER BY YEAR;</pre>
2)	<pre>-- 2 . Realice una consulta para mostrar el número de empleado, apellido y -- salario de todos los empleados que ganan más que el salario promedio y que -- trabajan en un departamento con cualquier empleado cuyo apellido contenga -- una "u" SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID, SALARY FROM EMPLOYEES E1 WHERE LAST_NAME LIKE '%u%' AND SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES E2 WHERE E1.DEPARTMENT_ID = E2.DEPARTMENT_ID);</pre>
3)	<pre>-- 3 . Cree un reporte para mostrar el manager id y el salario del empleado con -- el salario más bajo para ese jefe. Excluir a cualquiera cuyo jefe no sea -- conocido. Excluya cualquier grupo donde el salario mínimo sea de \$ 6,000 o -- menos. Ordene la salida en orden descendente de salario. SELECT MANAGER_ID, MIN_SALARY FROM (SELECT MANAGER_ID, MIN(SALARY) AS MIN_SALARY FROM EMPLOYEES WHERE MANAGER_ID IS NOT NULL GROUP BY MANAGER_ID HAVING MIN(SALARY) > 6000) ORDER BY MIN_SALARY DESC;</pre>
4)	<pre>-- 4 Genere una estadística en la que se muestre el total de empleados según el -- número de años que lleva trabajando en la empresa. Los resultados deben -- aparecer ordenados desde lo que los que llevan más años. SELECT EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM HIRE_DATE) AS YEARS_WORKING, COUNT(*) AS TOTAL_EMPLOYEES FROM EMPLOYEES GROUP BY EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM HIRE_DATE) ORDER BY YEARS_WORKING DESC;</pre>

5)	<p>-- 5 Escriba una consulta que muestre la cantidad de cargos o trabajos que se ejercen en cada país. Mostrar el nombre del país. Mostrar los países en orden alfabético</p> <pre> SELECT C.COUNTRY_NAME COUNT(E.JOB_ID) AS JOB_COUNT FROM JOBS J JOIN employees e ON j.job_id = e.job_id JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID JOIN LOCATIONS L ON D.LOCATION_ID = L.LOCATION_ID JOIN COUNTRIES C ON L.COUNTRY_ID = C.COUNTRY_ID GROUP BY C.COUNTRY_NAME ORDER BY C.COUNTRY_NAME;</pre>
6)	<p>-- 6 Por cada trabajo mostrar el monto total de salario que se paga a todos los empleados que cumplen ese trabajo, y salario total de los que trabajan en LONDRES, OXFORD Y SEATTLE. Mostrar el nombre completo del trabajo. Y que aparezcan primero aquellos trabajos con mayor masa salarial total.</p> <pre> SELECT J.JOB_TITLE, SUM(E.SALARY) AS TOTAL_SALARY_ALL, SUM(CASE WHEN L.CITY IN ('London') THEN E.SALARY ELSE 0 END) AS LONDON, SUM(CASE WHEN L.CITY IN ('Oxford') THEN E.SALARY ELSE 0 END) AS OXFORD, SUM(CASE WHEN L.CITY IN ('Seattle') THEN E.SALARY ELSE 0 END) AS SEATTLE FROM EMPLOYEES E JOIN JOBS J ON E.JOB_ID = J.JOB_ID JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID JOIN LOCATIONS L ON D.LOCATION_ID = L.LOCATION_ID GROUP BY J.JOB_TITLE ORDER BY TOTAL_SALARY_ALL DESC;</pre>
7)	<p>-- 7 Cuántos empleados ingresaron desde el año siguiente en que ingresó el último programador (Programmer)?</p> <pre> SELECT COUNT(*) AS EMPLOYEES_COUNT FROM EMPLOYEES WHERE HIRE_DATE > (SELECT MAX(HIRE_DATE) FROM EMPLOYEES WHERE JOB_ID = 'Programmer');</pre>
8)	<p>-- 8 Mostrar el total de empleados según los siguientes rangos de salarios: de menos de 4000, de 4000 a 10000 y 10000 o más</p> <pre> SELECT SUM(CASE WHEN SALARY < 4000 THEN 1 ELSE 0 END) AS SALARY_LESS_THAN_4000, SUM(CASE WHEN SALARY BETWEEN 4000 AND 10000 THEN 1 ELSE 0 END) AS SALARY_4000_TO_10000, SUM(CASE WHEN SALARY > 10000 THEN 1 ELSE 0 END) AS SALARY_MORE_THAN_10000 FROM EMPLOYEES;</pre>
9)	<p>-- 9 Escriba una consulta que liste los empleados que ganan más que el promedio de todos los empleados.</p> <pre> SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY FROM EMPLOYEES WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES);</pre>

10)	<pre>-- 10 Escriba una consulta que liste los empleados que ganan más que el -- promedio de todos los empleados de su departamento. SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, DEPARTMENT_ID, SALARY FROM EMPLOYEES E1 WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES E2 WHERE E1.DEPARTMENT_ID = E2.DEPARTMENT_ID) ORDER BY DEPARTMENT_ID, SALARY DESC;</pre>
PARTE 2: Reportes estadísticos sobre esquema "Proyectos2020"	
11)	<pre>-- 11. Visualizar la suma de sueldos de cada oficio de la oficina 'VENTAS' SELECT OFICIO,SUM(SUELDO) AS "SUELDO TOTAL" FROM FUNCIONARIOS WHERE IDOFI = (SELECT IDOFI FROM OFICINAS WHERE NOMBREOFI = 'VENTAS') GROUP BY OFICIO;</pre>
12)	<pre>-- 12. Visualizar las oficinas con más de 4 funcionarios. SELECT NOMBREOFI FROM OFICINAS O JOIN FUNCIONARIOS F ON O.IDOFI = F.IDOFI GROUP BY NOMBREOFI HAVING COUNT(F.IDFUNC) > 4;</pre>
13)	<pre>-- 13. Mostrar el apellido, oficio y sueldo de los funcionarios cuyo sueldo sea -- mayor que la media de todos -- los sueldos. La lista debe ordenarse por apellido SELECT APELLIDO, OFICIO, SUELDO FROM FUNCIONARIOS WHERE SUELDO > (SELECT AVG(SUELDO) FROM FUNCIONARIOS) ORDER BY APELLIDO;</pre>
14)	<pre>-- 14. ¿Cuántos oficios hay en cada oficina y cual es la media del sueldo anual -- de cada una? SELECT O.IDOFI, O.NOMBREOFI,COUNT(DISTINCT F.OFICIO) AS OFICIOS, ROUND(AVG(F.SUELDO) * 12, 2) AS "SUELDO ANUAL PROMEDIO" FROM OFICINAS O JOIN FUNCIONARIOS F ON O.IDOFI = F.IDOFI GROUP BY O.IDOFI, O.NOMBREOFI;</pre>
15)	<pre>-- 15.¿En que meses la empresa debe celebrar cumpleaños de sus funcionarios, -- cuántos cumpleaños hay por mes? SELECT TO_CHAR(FECHANAC, 'Mon') AS Mes,COUNT(*) AS Cumpleaños FROM FUNCIONARIOS GROUP BY TO_CHAR(FECHANAC, 'Mon') ORDER BY TO_DATE(TO_CHAR(FECHANAC, 'Mon'), 'Mon');</pre>

16)	<pre>-- 16. Obtener la información en la que se reflejen los apellidos, oficios y -- sueldos tanto de los funcionarios que superan el sueldo de Aguirre, como del -- propio Aguirre. Debe listarse ordenado por el sueldo de menor a mayor. SELECT APELLIDO,OFICIO, SUELDO FROM FUNCIONARIOS WHERE SUELDO > (SELECT SUELDO FROM FUNCIONARIOS WHERE APELLIDO = 'AGUIRRE') UNION SELECT APELLIDO,OFICIO,SUELDO FROM FUNCIONARIOS WHERE APELLIDO = 'AGUIRRE' ORDER BY SUELDO;</pre>
17)	<pre>-- 17 Obtener el total de funcionarios y de esos, cuantos ganan por arriba del -- promedio y cuantos ganan por debajo del promedio. WITH AvgSalary AS (SELECT AVG(SUELDO) AS AVG_SAL FROM FUNCIONARIOS) SELECT COUNT(*) AS "Total funcionarios", SUM(CASE WHEN SUELDO > (SELECT AVG_SAL FROM AvgSalary) THEN 1 ELSE 0 END) AS "Con sueldo mayor al promedio", SUM(CASE WHEN SUELDO <= (SELECT AVG_SAL FROM AvgSalary) THEN 1 ELSE 0 END) AS "Con sueldo menor al promedio" FROM FUNCIONARIOS;</pre>
18)	<pre>-- 18. Presentar un listado donde se muestren el código y apellido de cada -- supervisor, junto al número de funcionarios que supervisa directamente. -- Nótese que puede haber funcionarios que no tengan supervisados, en cuyo caso -- el número de supervisados debería ser cero. SELECT F_SUP.IDFUNC AS "IDFUNC", F_SUP.APELLIDO AS "APELLIDO", NVL(COUNT(F_SUB.IDFUNC), 0) AS "SUPERVISADOS" FROM FUNCIONARIOS F_SUP LEFT JOIN FUNCIONARIOS F_SUB ON F_SUP.IDFUNC = F_SUB.SUPERVISOR GROUP BY F_SUP.IDFUNC, F_SUP.APELLIDO ORDER BY F_SUP.IDFUNC;</pre>
19)	<pre>SELECT O.IDOFI, O.NOMBREOFI, SUM(F.SUELDO) AS "SUELDO TOTAL" FROM OFICINAS O JOIN FUNCIONARIOS F ON O.IDOFI = F.IDOFI GROUP BY O.IDOFI, O.NOMBREOFI HAVING SUM(F.SUELDO) = (SELECT MAX(SUM_SUELDO) FROM (SELECT SUM(F.SUELDO) AS SUM_SUELDO FROM OFICINAS O JOIN FUNCIONARIOS F ON O.IDOFI = F.IDOFI GROUP BY O.IDOFI));</pre>

20)	<pre>-- 20 listar los funcionarios que participan en proyectos y que fueron -- contratados después que Vivanco. En la información a mostrar del funcionario -- incluir la fecha de contrato en formato "lunes, 14 de Julio del 2013" SELECT F.IDFUNC, F.APELLIDO, TO_CHAR(F.FECHACONTRATO, 'Day, DD "de" Month "del" YYYY', 'NLS_DATE_LANGUAGE = Spanish') AS "FECHA DE CONTRATO" FROM FUNCIONARIOS F JOIN PARTICIPANTES P ON F.IDFUNC = P.IDFUNC WHERE F.FECHACONTRATO > (SELECT FECHACONTRATO FROM FUNCIONARIOS WHERE APELLIDO = 'VIVANCO') ORDER BY F.FECHACONTRATO;</pre>
21)	<pre>-- 21 Obtener los funcionarios con sueldo mayor de cada oficina. Se debe -- mostrar -- el nombre de la oficina, el funcionario con el sueldo más alto y su sueldo. -- Ordenar por nombre de oficina. SELECT o.NOMBREOFI, f.APELLIDO, f.SUELDO FROM FUNCIONARIOS f JOIN OFICINAS o ON f.IDOFI = o.IDOFI JOIN (SELECT IDOFI, MAX(SUELDO) AS MAX_SAL FROM FUNCIONARIOS GROUP BY IDOFI) m ON f.IDOFI = m.IDOFI AND f.SUELDO = m.MAX_SAL ORDER BY o.NOMBREOFI;</pre>
22)	<pre>-- 22 Para cada oficina con al menos dos funcionarios y tal que la media del -- sueldo sea mayor que la media de los sueldos de todos los funcionarios, -- muéstrese el código, el nombre de la oficina y la suma del sueldo de sus -- funcionarios. SELECT COUNT(*) AS "Total funcionarios", SUM(CASE WHEN SUELDO > (SELECT AVG(SUELDO) FROM FUNCIONARIOS) THEN 1 ELSE 0 END) AS "Con sueldo mayor al promedio", SUM(CASE WHEN SUELDO <= (SELECT AVG(SUELDO) FROM FUNCIONARIOS) THEN 1 ELSE 0 END) AS "Con sueldo menor al promedio" FROM FUNCIONARIOS;</pre>
PARTE 3: Implementación de código almacenado PL/SQL para esquema "Proyectos2020"	

23)	<pre>-- 23. Escriba una función que reciba como parámetro el código de un proyecto -- y devuelva el tipo de proyecto que corresponde. Los tipos de proyecto se -- detallan en la siguiente tabla: CREATE OR REPLACE FUNCTION TipoProyecto(p_IDPROY IN NUMBER) RETURN VARCHAR2 IS v_tipo_proyecto VARCHAR2(30); v_count_presidente NUMBER; v_count_director NUMBER; BEGIN -- Verificar si participa el PRESIDENTE SELECT COUNT(*) INTO v_count_presidente FROM PARTICIPANTES P JOIN FUNCIONARIOS F ON P.IDFUNC = F.IDFUNC WHERE P.IDPROY = p_IDPROY AND F.OFICIO = 'PRESIDENTE'; IF v_count_presidente > 0 THEN v_tipo_proyecto := 'Proyecto Estratégico'; ELSE -- Verificar si participa algún DIRECTOR SELECT COUNT(*) INTO v_count_director FROM PARTICIPANTES P JOIN FUNCIONARIOS F ON P.IDFUNC = F.IDFUNC WHERE P.IDPROY = p_IDPROY AND F.OFICIO = 'DIRECTOR'; IF v_count_director > 0 THEN v_tipo_proyecto := 'Proyecto Administrativo'; ELSE v_tipo_proyecto := 'Proyecto Operativo'; END IF; END IF; RETURN v_tipo_proyecto; END; /</pre>
24)	<pre>-- 24 Use la función creada en el punto anterior para generar un reporte que -- muestre por cada proyecto el nombre de proyecto, el tipo de proyecto y la -- cantidad de participantes -- Consulta para generar el reporte SELECT P.NOMBREPROY AS Proyecto, TipoProyecto(P.IDPROY) AS "Tipo de proyecto", COUNT(PAR.IDFUNC) AS Participantes FROM PROYECTOS P LEFT JOIN PARTICIPANTES PAR ON P.IDPROY = PAR.IDPROY GROUP BY P.IDPROY, P.NOMBREPROY ORDER BY P.NOMBREPROY;</pre>

25)

```
-- 25 . Deseamos que cuando se agregue una nueva participación en proyectos,
-- se verifique que la carga horaria semanal acumulada del funcionario no
supere
-- las 40 horas. Para ello se debe implementar un trigger en la tabla
-- PARTICIPANTES tal que cuando vaya a insertar un nuevo registro (una nueva
-- asignación de un funcionario a un proyecto), verifique que la suma de horas
-- del funcionario no sea mayor a 40. Si lo es, debería generar una excepcion
-- con el mensaje ''El total de carga horaria en proyectos supera las 40
horas'.

CREATE OR REPLACE TRIGGER trg_check_hours
BEFORE INSERT ON PARTICIPANTES
FOR EACH ROW
DECLARE
    v_total_hours NUMBER;
BEGIN
    -- Calcular la suma de horas actuales del funcionario en otros proyectos
    SELECT SUM(HORASSEMANA)
    INTO v_total_hours
    FROM PARTICIPANTES
    WHERE IDFUNC = :NEW.IDFUNC;

    -- Si la suma de horas actuales más las nuevas horas supera 40, lanzar una
    excepción
    IF v_total_hours + :NEW.HORASSEMANA > 40 THEN
        RAISE_APPLICATION_ERROR(-20001, 'El total de carga horaria en proyectos
        supera las 40 horas');
    END IF;
END;
/
```