

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL RESISTENCIA

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**BASES DE DATOS**

Guía de Trabajos Prácticos: Unidad 5

Carpeta de Entrega

**Grupo 4**

Integrantes:

* **Andres**, Aldo Omar.
* **Bravo Pérez**, Agustín Nicolás.
* **Brites,** Agustín.
* **Sáez Franci**, Juliana Carla Desiree.
* **Salomón**, Hilel Mauricio.
* **Machuain**, Ezequiel.

*10 de Junio de 2023*

***1. Dado el siguiente esquema relacional:***

***Proveedores(sid: integer, sname: string, address: string)***

***Parts(pid: integer, pname: string, color: string)***

***Catalog(sid: integer, pid: integer, cost: real)***

***1. Encuentre los snombres de Proveedores que provean alguna parte roja.***

SELECT sname FROM suppliers s

JOIN catalog c ON c.sid = s.sid

JOIN parts p ON p.pid = c.pid

WHERE p.color = 'red';

***2. Encuentre los sids de Proveedores que provean alguna red o parte verde.***

SELECT sid FROM catalog c

JOIN parts p ON p.pid = c.pid

WHERE p.color = 'red' OR p.color = 'green';

***3. Encuentre los sids de Proveedores que provean alguna parte roja o vivan en “221 Packer Street”.***

SELECT s.sid FROM suppliers s

JOIN catalog c ON c.sid = s.sid

JOIN parts p ON p.pid = c.pid

WHERE p.color = 'red' OR s.address = '221 Packer Street';

***4. Encuentre los sids de Proveedores que provean alguna parte roja and alguna parte verde.***

SELECT sid FROM catalog c

JOIN parts p ON p.pid = c.pid

WHERE p.color = 'red' AND sid IN (SELECT sid FROM catalog c

JOIN parts p ON p.pid = c.pid

WHERE p.color = 'green');

***5. Encuentre los sids de Proveedores que provean cada parte.***

SELECT c.sid FROM catalog c

WHERE NOT EXISTS (SELECT p.pid FROM parts p

WHERE p.pid NOT IN (SELECT c2.pid FROM catalog c2

WHERE c2.sid = c.sid));

***6. Encuentre los sids de Proveedores que provean cada parte roja.***

SELECT c.sid FROM catalog c

WHERE NOT EXISTS (SELECT p.pid FROM parts p

WHERE p.color = 'red'

AND p.pid NOT IN (SELECT c2.pid FROM catalog c2

WHERE c2.sid = c.sid));

***7. Encuentre los sids de Proveedores que provean cada parte verde o roja.***

SELECT c.sid FROM catalog c

WHERE NOT EXISTS (SELECT p.pid FROM parts p

WHERE (p.color = 'red' OR p.color = 'green')

AND p.pid NOT IN (SELECT c2.pid FROM catalog c2

WHERE c2.sid = c.sid));

***8. Encuentre los sids de Proveedores que provean cada parte roja o provean cada parte verde.***

SELECT c.sid FROM catalog c

WHERE NOT EXISTS (SELECT p.pid FROM parts p

WHERE p.color = 'red'

AND p.pid NOT IN (SELECT c2.pid FROM catalog c2

WHERE c2.sid = c.sid))

UNION

SELECT c.sid FROM catalog c

WHERE NOT EXISTS (SELECT p.pid FROM parts p

WHERE p.color = 'green'

AND p.pid NOT IN (SELECT c2.pid FROM catalog c2

WHERE c2.sid = c.sid));

***9. Encuentre los pares de sids tal que los Proveedores con el primer sid cueste mas por alguna parte que los proveedor del segundo sid.***

SELECT c1.sid, c2.sid FROM catalog c1, catalog c2

WHERE c1.pid = c2.pid AND c1.cost > c2.cost

***10. Encuentre los pids de partes provista por al menos dos proveedores diferentes.***

SELECT DISTINCT c.pid FROM catalog c, catalog c2

WHERE c.pid = c2.pid AND c.sid <> c2.sid

***11. Encuentre los pids de las aprtes mas caras provistas por el proveedor llamado Yosemite Sham.***

SELECT c.pid FROM catalog c

JOIN suppliers s ON s.sid = c.sid

WHERE s.sname = 'Yosemite Sham'

AND c.cost = (SELECT MAX(c2.cost) FROM catalog c2

WHERE c.sid = c2.sid);

***12. Encuentre los pids de partes provistas por cada proveedor a menos que $200. (Si algún proveedor no provee las partes o cuesta igual o más de $200, las partes no deberán ser listadas.)***

SELECT p.pid FROM parts p

WHERE NOT EXISTS (SELECT s.sid FROM suppliers s

WHERE s.sid NOT IN (SELECT c.sid FROM catalog c

WHERE c.pid = p.pid

AND c.cost < 200));

***2. Dado el siguiente esquema relacional:***

***Vuelos(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time)***

***Avión(aid: integer, aname: string, cruisingrange: integer)***

***Certificados(eid: integer, aid: integer)***

***Empleados(eid: integer, ename: string, salary: integer)***

***a) Encuentre los eids de pilotos certificados para algún avión Boeing.***

SELECT DISTINCT c.eid FROM certified c

NATURAL JOIN aircraft a

WHERE a.aname LIKE '%Boeing%';

***b) Encuentre los nombres de pilotos certificados para algún avión Boeing.***

SELECT DISTINCT ename FROM employees e

NATURAL JOIN certified c

NATURAL JOIN aircraft a

WHERE aname LIKE '%Boeing%';

***c) Encuentre los aids de todos los aviones que pueden ser usados para vuelos sin parada desde Bonn hasta Madras.***

SELECT a.aid FROM aircraft a, flights f

WHERE a.crusingrange > f.distance

AND f.origin = 'Bonn'

AND f.destination = 'Madras';

***d) Identifique los vuelos que pueden ser piloteados por cada pilot cuyo salario sea mayor a $100,000.***

SELECT DISTINCT f.flno FROM flights f, aircraft a

NATURAL JOIN certified c

NATURAL JOIN employees e

WHERE e.salary > 100000

AND a.crusingrange > f.distance;

***e) Encuentre los nombres de pilotos que pueden operar aviones con un rango mayor a 3,000 millas pero que NO esten certificados para los aviones Boeing.***

SELECT e.ename FROM employees e

NATURAL JOIN certified c

NATURAL JOIN aircraft a

WHERE a.crusingrange > 3000

AND e.eid NOT IN (SELECT c2.eid FROM certified c2

NATURAL JOIN aircraft a2

WHERE a2.aname LIKE '%Boeing%');

***f) Encuentre los eids de empleados que ganan el mayor salario.***

SELECT e.eid FROM employees e

WHERE e.salary = (SELECT MAX(salary) FROM employees);

***g) Encuentre los eids de empleados que ganen el segundo mayor salario.***

SELECT e.eid FROM employees e

WHERE e.salary = (SELECT MAX(salary) FROM employees

WHERE salary <> (SELECT MAX(salary)

FROM employees));

***h) Encuentre los eids de empleados que están certificados para el mayor número de aviones.***

SELECT MAX(count) FROM (SELECT COUNT(c2.eid) count

FROM certified c2

GROUP BY c2.eid) AS counts;

***i) Encuentre los eids de los empleados que están certificados para exactamente 3 (tres) aviones.***

SELECT eid FROM certified c

GROUP BY eid

HAVING COUNT(eid) = 3;

***j) Encuentre la cantidad total de dinero pagado en concepto de salario.***

SELECT SUM(salary) FROM employees;

***3. Dado el siguiente esquema relacional:***

***Student(snum: integer, sname: string, major: string, level: string, age: integer)***

***Class(name: string, meets at: string, room: string, fid: integer)***

***Inscripto(snum: integer, cname: string)***

***Faculty(fid: integer, fname: string, deptid: integer)***

***a) Encuentre los nombres de todos los estudiantes Juniors (level = JR) que esten enrolados en una clase dictada por I. Teach.***

SELECT DISTINCT s.sname FROM student s

NATURAL JOIN enrolled

NATURAL JOIN class

NATURAL JOIN faculty f

WHERE s.standing = 'JR' AND f.fname LIKE 'I. Teach';

***b) Encuentre los nombres de todas clases que cumpla estar en el aula R128 o tenga mas de 5 estudiantes inscriptos.***

SELECT c.name FROM class c

WHERE c.room = "R128"

UNION

SELECT c2.name FROM class c2 INNER JOIN enrolled e ON e.cname=c2.name

GROUP BY c2.name

HAVING COUNT(c2.name) > 5

***c) Encuentre los nombres de todos estudiantes que estan inscriptos en dos clases que comiencen a la mismo hora.***

SELECT s.sname FROM student s

WHERE s.snum IN (

SELECT e1.snum FROM enrolled e1, class c1, enrolled e2, class c2

WHERE e1.snum = e2.snum

AND e1.cname = c1.name AND e2.cname = c2.name

AND c1.name <> c2.name AND c1.meets\_at = c2.meets\_at

);

***d) Encuentre los nombres de miembros de la facultad que hay en cada aula en la cual se dicta alguna clase.***

SELECT f.fname FROM faculty f

WHERE NOT EXISTS (SELECT c.room FROM class c

WHERE c.room NOT IN (SELECT c2.room FROM class c2

WHERE c2.fid = f.fid));

***e) Encuentre los nombres de los miembros de la facultad para quienes la inscripción combinada de los cursos que enseñan es menos de cinco.***

SELECT fname FROM faculty f

WHERE 5 > (SELECT COUNT(e.snum) FROM class c

INNER JOIN enrolled e ON e.cname=c.name

WHERE c.fid = f.fid);

***f) Para cada nivel, liste los niveles y los promedios de edad de los estudiantes para ese nivel.***

SELECT standing, AVG(age) FROM student

GROUP BY standing

***g) Para todos los niveles excepto JR, liste el nivel y los promedios de edad de estudiantes para dicho nivel.***

SELECT standing, AVG(age) FROM student

WHERE standing <> 'JR'

GROUP BY standing;

***h) Para cada miembro de la facultad que toma clases en el aula R128, liste los nombres de los miembros de la facultad y el total de clases que toma.***

SELECT f.fname, COUNT(c.name)

FROM faculty f

INNER JOIN class c ON f.fid = c.fid

WHERE c.room = 'R128'

GROUP BY f.fid;

***i) Encuentre los nombres de estudiantes inscriptos en para el máximo nro. de clases.***

SELECT s.sname FROM student s

INNER JOIN enrolled e ON e.snum = s.snum

GROUP BY e.snum

HAVING COUNT(e.snum) = (

SELECT MAX(nroCursos)

FROM (

SELECT COUNT(e2.snum) AS nroCursos

FROM enrolled e2

GROUP BY e2.snum) AS cursosPorEstudiante

);

***j) Encuentre los nombres de estudiantes no inscriptos in ninguna clase.***

SELECT DISTINCT s.snum FROM student s

WHERE NOT EXISTS (

SELECT \* FROM enrolled e

WHERE s.snum = e.snum

);

-- Alternativa

SELECT s.snum FROM student s

LEFT OUTER JOIN enrolled e ON e.snum = s.snum

WHERE e.snum IS NULL;

***k) Para cada valor de la edad que aparezca en estudiantes, encuentre el valor del nivel que aparece lo más a menudo posible. Por ejemplo, si hay más estudiantes del nivel del FR con 18 años que en los niveles SR, JR, o SO, usted debe imprimir el par (18, FR).***

SELECT s.age, s.standing FROM student s

GROUP BY s.age, s.standing

HAVING COUNT(s.standing) = (

SELECT MAX(standingCount)

FROM (

SELECT COUNT(s2.standing) AS standingCount

FROM student s2 where s2.age = s.age

GROUP BY s2.age, s2.standing

) AS standingsPorEdad

);

***4. Dado el siguiente esquema relacional:***

***Emp(eid: integer, ename: string, age: integer, salary: real)***

***Works(eid: integer, did: integer, pct time: integer)***

***Dept(did: integer, dname: string, budget: real, managerid: integer)***

***a) Liste los nombres las edades de cada employee que trabajan en los departamentos de Hardware y los departmento de Software.***

SELECT ename, age FROM emp e

NATURAL JOIN works w

NATURAL JOIN dept d

WHERE d.dname IN ('Hardware', 'Software');

***b) Encuentre los managerids de los gerentes que administran solo departmentos con presupuesto mayor a $1.000.000.***

SELECT managerid FROM dept

WHERE managerid NOT IN (SELECT d2.managerid FROM dept d2

WHERE d2.budget <= 1000000);

***c) Encuentre los enames de los gerentes que administran los departamentos con los mayores presupuestos. Si un gerente administra más de un depto. se debe sumar sus presupuestos.***

SELECT e.ename FROM emp e

INNER JOIN dept d ON d.managerid = e.eid

GROUP BY d.managerid

HAVING SUM(d.budget) = (

SELECT MAX (budget)

FROM (

SELECT SUM(d2.budget) AS budget

FROM dept d2

GROUP BY d2.managerid) AS budgets

)

);

***d) Encuentre los managerids de los genretes que controlan mas de $5.000.000.***

SELECT managerid FROM dept

GROUP BY (managerid)

HAVING SUM(budget) > 5000000;

***e) Encuentre los managerids de los gerentes que controlan los mayores presupuestos.***

SELECT d.managerid FROM dept d

GROUP BY (d.managerid)

HAVING SUM(d.budget) = (

SELECT MAX (budget)

FROM (

SELECT SUM(d2.budget) AS budget

FROM dept d2

GROUP BY d2.managerid) AS budgetsPerManager

)

);

***f) Encuentre los enames de los gerentes que administran solo departmentos con presupuestos de más de $1.000.000, pero al menos un depto. tiene presupuesto menor a $5.000.000.***

SELECT e.ename FROM emp e

INNER JOIN dept d ON d.managerid = e.eid

WHERE d.budget < 5000000

AND d.managerid NOT IN (

SELECT d2.managerid FROM dept d2

WHERE d2.budget <= 1000000

);