**BASE DE DATOS RELACIONAL CONSULTAS:**

**RUBRICA:**

**Cada miembro del equipo crea tres (3) consultas para la base de datos relacional, considerando una (1) básica y dos (2) de nivel intermedio que incluyan la creación de procedimientos almacenados y/o funciones, agregaciones, joins, subconsultas. Para cada una de las consultas se indica el propósito de la misma.**

**FRANCO FELIX YANCE GUTIERREZ:**

**CONSULTA 01:**

**-- Esta consulta selecciona todos los datos de la tabla 'usuarios'**

**SELECT \* FROM usuarios;**

**CONSULTA 02:**

**-- Esta consulta calcula el salario promedio de las ofertas de trabajo para cada empresa**

**-- Primero, une las tablas 'empresas', 'oferta\_por\_empresa' y 'ofertas' basándose en los códigos correspondientes**

**-- Luego, agrupa los resultados por el nombre de la empresa**

**-- Finalmente, calcula el salario promedio para cada grupo (cada empresa)**

**SELECT e.nombre, AVG(o.sueldo) as salario\_promedio**

**FROM empresas e**

**JOIN oferta\_por\_empresa oe ON e.codigo = oe.empresas\_codigo**

**JOIN ofertas o ON oe.ofertas\_codigo = o.codigo**

**GROUP BY e.nombre;**

**CONSULTA 03:**

**-- Este procedimiento almacenado recupera todas las ofertas de trabajo para una empresa dada**

**-- El nombre de la empresa se pasa como parámetro (@nombreEmpresa)**

**-- Primero, une las tablas 'empresas', 'oferta\_por\_empresa' y 'ofertas' basándose en los códigos correspondientes**

**-- Luego, selecciona todas las ofertas de trabajo donde el nombre de la empresa coincide con el parámetro dado**

**CREATE PROCEDURE ObtenerOfertasPorEmpresa(@nombreEmpresa VARCHAR(50))**

**AS**

**BEGIN**

**SELECT o.\***

**FROM empresas e**

**JOIN oferta\_por\_empresa oe ON e.codigo = oe.empresas\_codigo**

**JOIN ofertas o ON oe.ofertas\_codigo = o.codigo**

**WHERE e.nombre = @nombreEmpresa;**

**END;**

**-- Este comando ejecuta el procedimiento almacenado 'ObtenerOfertasPorEmpresa'**

**-- Pasamos 'Microsoft' como el parámetro @nombreEmpresa**

**EXEC ObtenerOfertasPorEmpresa @nombreEmpresa = 'Microsoft';**

**PEDRO ALEJANDRO RETUERTO DIAZ**

**–muestra el nombre del campo de trabajo con sueldo mayor a 1000, acorde a un parametro de funcion**

CREATE FUNCTION CampoSueldo(@nombre varchar(50)) RETURNS TABLE

AS

RETURN

SELECT p.campo, o.sueldo

FROM ofertas AS o

JOIN postulaciones AS p ON p.codigo = o.codigo

WHERE o.sueldo > 1000

AND p.campo LIKE '%' + @nombre + '%';

go;

select \* from CampoSueldo('a')

—------------------------------------------------------------------------------------------

–crea una vista para visualizar el nombre de pais, codigo y nombre

CREATE VIEW MostrarNombrePaisCiudad AS

SELECT p.codigo, p.nombre AS nombre\_pais, c.nombre AS nombre\_ciudad

FROM paises AS p

JOIN ciudades AS c ON p.codigo = c.paises\_codigo;

go;

select \* from MostrarNombrePaisCiudad

—-------------------------------------------------------------------------------------------

–selecciona las postulaciones que tengan como contenido inicial la palabra desarroll

select \* from postulaciones

where campo like'desarroll%'

go;

—--------------------------------------------------------------------------------------------

**Cesar Guerrero:**

-- **Mostrar la ciudad con más instituciones**

CREATE VIEW Cantidad\_de\_instituciones\_por\_Ciudad

as

SELECT c.nombre, COUNT(DISTINCT(i.nombre)) as 'Cantidad'

FROM ciudades as c

JOIN instituciones as i

ON c.codigo = i.ciudades\_codigo

GROUP BY c.nombre

SELECT c.nombre

FROM ciudades as c

JOIN instituciones as i

ON c.codigo = i.ciudades\_codigo

GROUP BY c.nombre

HAVING COUNT(DISTINCT(i.nombre)) =

(SELECT MAX(Cantidad)

FROM Cantidad\_de\_instituciones\_por\_Ciudad as Temporal)

**-- Mostrar los nombres de las personas que hayan tenido algun estudio o curso en la UPC.**

CREATE PROCEDURE usp\_print\_usuarios\_en\_UPC

as

begin

declare @nombre varchar(100)

declare cursor\_name cursor for

SELECT DISTINCT CONCAT(u.nombre, ' ', u.apellido\_paterno)

FROM usuarios as u

JOIN curriculums as cu ON u.curriculums\_codigo = cu.codigo

JOIN curriculums\_por\_carreras as cpc ON cu.codigo = cpc.curriculums\_codigo

JOIN carreras as ca ON cpc.carreras\_codigo = ca.codigo

JOIN carreras\_por\_instituciones as cpi ON ca.codigo = cpi.estudios\_academicos\_codigo

JOIN instituciones as i ON cpi.instituciones\_codigo = i.codigo

JOIN curriculums\_por\_certificados as cpce ON cu.codigo = cpce.curriculums\_codigo

JOIN certificaciones as ce ON cpce.certificaciones\_codigo = ce.codigo

JOIN certificaciones\_por\_instituciones as cepi ON ce.codigo = cepi.certificaciones\_codigo

JOIN instituciones ON cepi.instituciones\_codigo = i.codigo

JOIN curriculums\_por\_est\_compl as cpec ON cu.codigo = cpec.curriculums\_codigo

JOIN estudios\_complementarios as ec ON cpec.est\_compl\_codigo = ec.codigo

JOIN instituciones as ins ON ec.codigo = ins.codigo

WHERE i.nombre = 'UPC'

open cursor\_name

fetch cursor\_name into @nombre

while(@@FETCH\_STATUS = 0)

begin

print(CONVERT(varchar(100), @nombre))

fetch cursor\_name into @nombre

end

close cursor\_name

deallocate cursor\_name

end

exec usp\_print\_usuarios\_en\_UPC

**-- Mostrar los nombres de las personas que tengan las certificacion de SCRUM y hablen ingles**

CREATE FUNCTION Usuarios\_con\_scrum\_ingles () returns table

as

return

SELECT CONCAT(u.nombre,' ', u.apellido\_paterno) as Nombre

FROM usuarios as u

JOIN curriculums as cu

ON u.curriculums\_codigo = cu.codigo

JOIN curriculums\_por\_certificados as cpce

ON cu.codigo = cpce.curriculums\_codigo

JOIN certificaciones as ce

ON cpce.certificaciones\_codigo = ce.codigo

JOIN curriculums\_por\_idiomas as cpi

ON cu.codigo = cpi.curriculums\_codigo

JOIN idiomas as idio

ON cpi.idiomas\_codigo = idio.codigo

WHERE ce.descripcion LIKE 'SCRUM' AND idio.descripcion = 'Ingles'

select \* from Usuarios\_con\_scrum\_ingles()

**Raizo Valdivia**

**Mostrar el nombre de las instituciones con su respectiva ciudad donde se ubica**

CREATE VIEW instituciones\_alfabetico\_por\_ciudades

AS

SELECT ciudades.nombre AS ciudad, instituciones.nombre AS institucion

FROM ciudades

JOIN instituciones ON ciudades.codigo = instituciones.ciudades\_codigo

SELECT \*

FROM instituciones\_alfabetico\_por\_ciudades

ORDER BY institucion DESC;

**Mostrar las ofertas que tiene cada empresa y de dónde son**

CREATE VIEW lista\_empresas\_ofertas\_paises AS

SELECT e.nombre AS empresa, o.puesto, p.nombre AS pais

FROM empresas e

JOIN empresas\_por\_ciudades ec ON e.codigo = ec.empresas\_codigo

JOIN ciudades c ON ec.ciudades\_codigo = c.codigo

JOIN paises p ON c.paises\_codigo = p.codigo

JOIN oferta\_por\_empresa oe ON e.codigo = oe.empresas\_codigo

JOIN ofertas o ON oe.ofertas\_codigo = o.codigo;

SELECT \* FROM lista\_empresas\_ofertas\_paises;

**Mostrar los datos de la tabla empresas, ordenados dependiendo de quien tenga el correo con más caracteres.**

CREATE VIEW empresas\_ordenadas AS

SELECT \*

FROM empresas

ORDER BY LENGTH(correo\_electronico) DESC;

SELECT \* FROM empresas\_ordenadas;

**ALdair Cruz:**

**-Mostrar el nombre de la empresa y consecuentemente nos muestre el respectivo país donde pertenece:**

CREATE VIEW empresas\_por\_paises AS

SELECT paises nombre AS pais, empresas nombre AS empresa

FROM paises

JOIN empresas ON paises.codigo = empresas.paises\_codigo

ORDER BY empresas.nombre

SELECT \* FROM empresas\_por\_paises

**-Mostrar el nombre de las carreras y la universidad a la cual pertenece**

CREATE VIEW carreras\_orden\_por\_instituciones AS

SELECT instituciones.nombre AS institucion, carreras.nombre AS carrera

FROM instituciones

JOIN carreras ON instituciones.codigo = carreras.instituciones\_codigo

ORDER BY carreras.nombre DESC;

SELECT \* FROM carreras\_orden\_por\_instituciones

**- Mostrar el nombre de los postulantes que tengan el AWS Certified y hablen y/o dominen el chino madarín**

CREATE FUNCTION Usuarios\_con\_aws\_chinom () RETURNS TABLE AS

RETURN

SELECT CONCAT(u.nombre,' ', u.apellido\_paterno) as Nombre

FROM usuarios AS usu

JOIN curriculums AS cur

ON usu.curriculums\_codigo = cur.codigo

JOIN curriculums\_por\_certificados AS cupoce

ON cur.codigo = cupoce.curriculums\_codigo

JOIN certificaciones AS cer

ON cupoce.certificaciones\_codigo = cer.codigo

JOIN curriculums\_por\_idiomas AS cupoid

ON cur.codigo = cupoid.curriculums\_codigo

JOIN idiomas AS idi

ON cupoid.idiomas\_codigo = idi.codigo

WHERE cer.descripcion LIKE 'AWS Certified' AND idio.descripcion = 'Chino Mandarin'

select \* from Usuarios\_con\_aws\_chinom()

Alex Espinoza

CONSULTA 1:

**Ver los nombres de los usuarios y las carreras que han estudiado**

CREATE VIEW vw\_UsuariosYCarreras

AS

SELECT u.nombre, ca.descripcion as carrera

FROM usuarios u

JOIN curriculums\_por\_carreras cpc ON u.curriculums\_codigo = cpc.curriculums\_codigo

JOIN carreras ca ON cpc.carreras\_codigo = ca.codigo;

SELECT \* FROM vw\_UsuariosYCarreras;

CONSULTA 2:

**Obtener el número total de usuarios que han estudiado cada carrera**

SELECT ca.descripcion, COUNT(u.codigo) as total\_usuarios

FROM usuarios u

JOIN curriculums\_por\_carreras cpc ON u.curriculums\_codigo = cpc.curriculums\_codigo

JOIN carreras ca ON cpc.carreras\_codigo = ca.codigo

GROUP BY ca.descripcion;

CONSULTA 3:

**Obtener una lista de usuarios que han estudiado una carrera específica**

CREATE PROCEDURE ObtenerUsuariosPorCarrera(@nombreCarrera VARCHAR(50))

AS

BEGIN

SELECT u.\*

FROM usuarios u

JOIN curriculums\_por\_carreras cpc ON u.curriculums\_codigo = cpc.curriculums\_codigo

JOIN carreras ca ON cpc.carreras\_codigo = ca.codigo

WHERE ca.descripcion = @nombreCarrera;

END;

EXEC ObtenerUsuariosPorCarrera @nombreCarrera = 'Ingeniería de Sistemas';