CREATE DATABASE `Prueba1`

CREATE TABLE `usuarios` (

`userID` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`Login` VARCHAR(100),

`NOMBRES` VARCHAR(100),

`PATERNO` VARCHAR(100),

`MATERNO` VARCHAR(100),

PRIMARY KEY (`userID`) USING BTREE

);

CREATE TABLE `empleados` (

`userId` INT UNSIGNED NOT NULL,

`Sueldo` DOUBLE UNSIGNED NULL DEFAULT NULL,

`Fecha Ingreso` DATE NULL DEFAULT NULL,

INDEX `FK\_empleados\_usuarios` (`userId`) USING BTREE,

CONSTRAINT `FK\_empleados\_usuarios` FOREIGN KEY (`userId`) REFERENCES `usuarios` (`userID`) ;

--Depurar solo los ID diferentes de 6,7,9 y 10 de la tabla usuarios (5 puntos)

USE PRUEBA1;

DELETE FROM usuarios

WHERE userId IN (6, 7, 9);

--Actualizar el dato Sueldo en un 10 porciento a los empleados que tienen fechas entre el año 2000 y 2001

UPDATE empleados

SET sueldo = sueldo \* 1.10 -- Aumento del 10%

WHERE FechaIngreso LIKE '2000%' OR FechaIngreso like '2001%';

--Realiza una consulta para traer el nombre de usuario y fecha de ingreso de los usuarios que gananen mas de 10000 y su apellido comience con T ordernado del mas reciente al mas antiguo

Select Nombres, FechaIngreso

FROM usuarios

INNER JOIN empleados

ON usuarios.userId = empleados.userId

WHERE empleados.sueldo > 10000

AND usuarios.PATERNO LIKE 'T%'

order by empleados.FechaIngreso desc;

--Realiza una consulta donde agrupes a los empleados por sueldo, un grupo con los que ganan menos de 1200 y uno mayor o igual a 1200, cuantos hay en cada grupo?

SELECT

CASE

WHEN sueldo < 1200 THEN 'Menos de 1200'

WHEN sueldo >= 1200 THEN '1200 o más'

END AS grupo\_sueldo,

COUNT(\*) AS cantidad\_empleados

FROM empleados

GROUP BY

CASE

WHEN sueldo < 1200 THEN 'Menos de 1200'

WHEN sueldo >= 1200 THEN '1200 o más'

END;