**Evaluación Continua 1**

**ADMINISTRACION DE BASE DE DATOS.**

**Descontar 2pts, si no se cumple este requerimiento.**

Apellidos y Nombres: CHAVES RAMOS MIGUEL ALFONZO

Sección: III.13.2023-IE Sede: SEDE VIRTUAL

**Pregunta 1: (1 Pts.)**

Mencione cuales son los servicios necesarios para levantar el motor de base de datos Oracle.

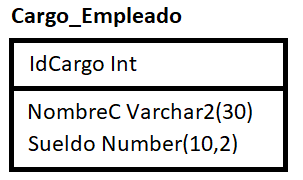
**Pregunta 2: (2 Pts.)**

Crear una cuenta de usuario con el nombre **EC1** y la contraseña de acceso es: 123456789, el cual está asociada con el archivo de datos **C:\IDAT\EVALUACION\_1.DBF**, cuyo tamaño es de **200** megabytes. Asigne los privilegios necesarios, Redacte el código correspondiente el cual permita esta acción. No olvide cerrar todas las ventanas y conexiones y probar la nueva cuenta de usuario. **Se requiere el código**.

**Nota:** DE AQUÍ EN ADELANTE LAS SIGUIENTES PREGUNTAS SE RESUELVEN USANDO EL NUEVO USUARIO.

**Pregunta 3: (3 Pts.)**

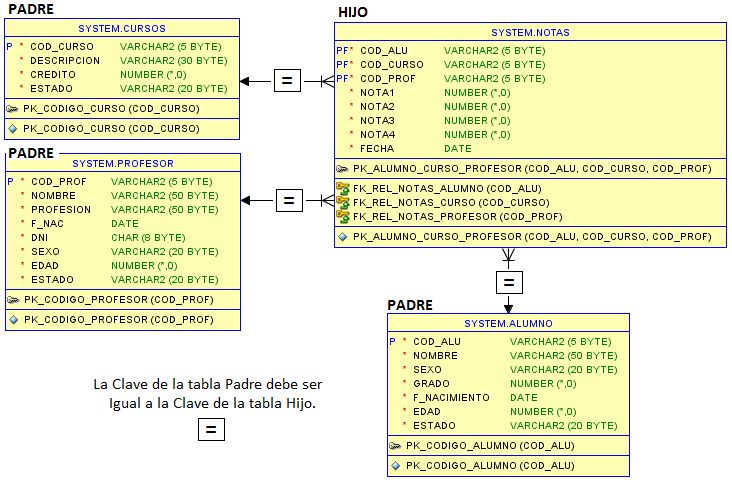
Crear una tabla con las siguientes características:



* Crear una secuencia el cual permita generar códigos de cargo en el campo **IdCargo**, esta se genera de 1 en 1 hasta el valor 10000. Realice las pruebas pertinentes.
* Crear un índice sobre la columna **NombreC**.
* Asigne a la tabla **Cargo\_Empleado** el sinónimo **CargoX**.

**Pregunta 4: (6 Pts.)**

Se presenta el siguiente modelo Relacional.



Realice el ***Script de base de datos*** del presente modelo, tome en cuenta lo siguiente:

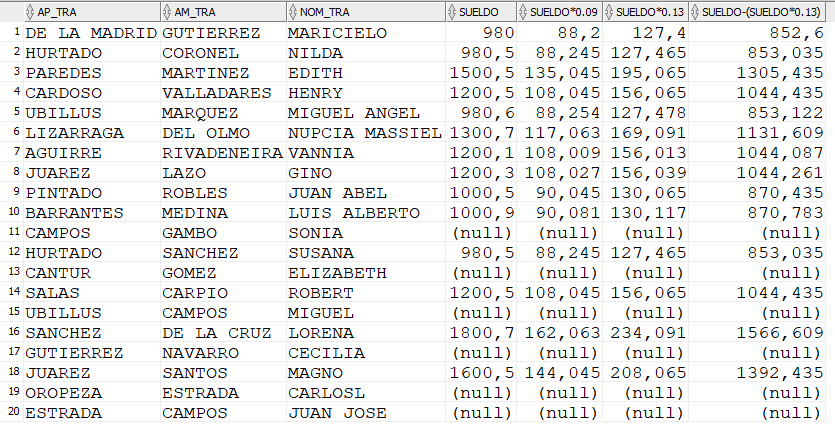
* Llaves Primarias
* Llaves Foráneas
* Relaciones
* Constructores para validar las columnas de cada tabla, dado que no se puede ingresar datos que no corresponden (Validación de datos)
* Al final muestre su diagrama de base de datos.

**Pregunta 5: (8 Pts.)**

El Presente examen viene con un Script de Base de datos con el nombre **EC1.sql**, se pide ejecutarlo y realizar las siguientes consultas **SELECT**:

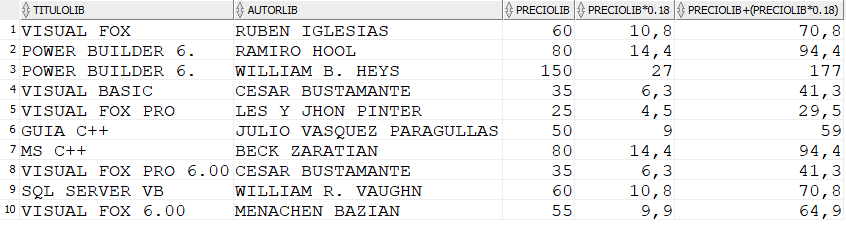
1. de la tabla **TRABAJADOR** obtener los siguientes resultados, considere que el trabajador tiene descuentos por **Essalud** (9%), **Onp** (13%) y cuanto se le debe abonar en su cuenta como pago mensual.

**Resultado:**



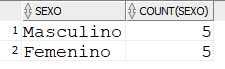
1. el dueño de la librería desea vender los libros del área **LPROG**, aquí él está considerando aplicar a sus precios el **IGV** correspondiente al **18%**, y además desea saber cuánto pagará el cliente, tome en cuenta que en la venta se incluye el IGV.

**Resultado:**



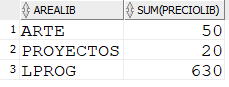
1. una escuela desea clasificar al alumnado por el género, indicando cuantos varones y mujeres existen en dicha escuela.

**Resultado:**



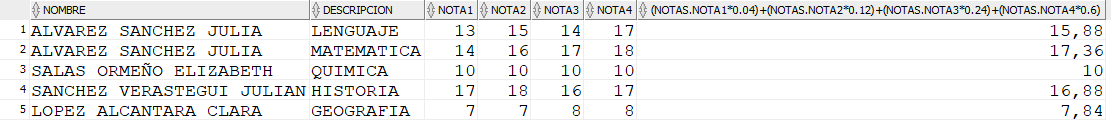
1. Se desea clasificar los libros de nuestra librería, por el área de los libros, indicando la suma de sus precios por cada área, mostrando únicamente en el reporte las clasificaciones **ARTE**, **PROYECTOS** y **LPROG**.

Resultado:



1. Se desea saber cuáles son los **profesores**, **cursos**, **calificaciones** incluyendo el **promedio** del alumno, en este caso del alumno **PEREZ SANCHEZ JUAN**. Resuelva esta pregunta de 2 maneras:
   1. Exclusivamente con la regla de igualdad de llaves
   2. Usando el NATURAL JOIN.

Resultado en ambos casos debe salir:



**SOLUCIÓN:**

Pegar aquí las imágenes o códigos que correspondan a la solución del presente examen de forma clara y ordenada.

**Pregunta 1:**  
1-. OracleServiceXE

2-. OracleXETNSListener

**Pregunta 2:**

--BORRANDO EL TABLESPACE, USUARIO Y ROL, SI EXISTIERAN

DROP ROLE ROL\_EX;

DROP USER EC1 CASCADE;

DROP TABLESPACE EV01 INCLUDING CONTENTS;

--1 Creando el TableSpace

Create TableSpace EV01 DataFile 'C:\IDAT\EVALUACION\_1.DBF' Size 200M;

--2 Verificando la creación del TableSpace

Select \* From v$tablespace;

--3 Ahora crearemos al usuario EC1 , cuya contraseña será: 123456789

Create User EC1 Identified by "123456789"

Default TableSpace EV01

Temporary TableSpace TEMP

Quota unlimited on EV01;

--4 Creamos un Rol el cual contendrá los permisos, en este caso

-- el Role tendrá el nombre ROL\_EX.

Create Role ROL\_EX;

--5 Asignaremos los permisos (GRANT) al rol creado.

Grant CONNECT, CREATE TABLE, RESOURCE,

ALTER ANY INDEX, ALTER ANY SEQUENCE, ALTER ANY TABLE,

ALTER ANY TRIGGER, CREATE ANY INDEX, CREATE ANY SEQUENCE,

CREATE ANY SYNONYM, CREATE ANY TABLE, CREATE ANY TRIGGER,

CREATE ANY VIEW, CREATE PROCEDURE, CREATE PUBLIC SYNONYM,

CREATE TRIGGER, CREATE VIEW, DROP ANY INDEX,

DROP ANY SEQUENCE, DROP ANY TABLE, DROP ANY TRIGGER, DROP ANY VIEW,

INSERT ANY TABLE, QUERY REWRITE, SELECT ANY TABLE,

UNLIMITED TABLESPACE To ROL\_EX;

--6 Asociamos (GRANT) el Rol PERMITIR4 con el usuario EC1,

--de esta manera el usuario EC1, podrá realizar las tareas

--i/o permisos que el Rol tiene.

Grant ROL\_EX To EC1;

--7. CERRAR TODO, DESCONECTAR E INICIAR NUEVAMENTE, EN ESTA OCASION USAR

-- AL USUARIO EC1.

**Pregunta 3:**

--PREGUNTA 03

Create Table Cargo\_Empleado(

IdCargo Int Not Null,

NombreC VARCHAR2(30) Not Null,

Sueldo Number(10,2) Not Null,

PRIMARY KEY (IdCargo)

);

INSERT INTO Cargo\_Empleado values ('1','Matias','1600');

INSERT INTO Cargo\_Empleado values ('2','Joseph','2250');

INSERT INTO Cargo\_Empleado values ('3','Angie','2320');

SELECT \* FROM Cargo\_Empleado;

CREATE SEQUENCE Cargo\_SEQC

INCREMENT BY 1

START WITH 5

MINVALUE 1

MAXVALUE 10000

NOCACHE

NOCYCLE;

--Probando la secuencia

INSERT INTO Cargo\_Empleado values (Cargo\_ SEQC.NEXTVAL,'Matias','1600');

INSERT INTO Cargo\_Empleado values (Cargo\_ SEQC.NEXTVAL,'Joseph','2250');

INSERT INTO Cargo\_Empleado values (Cargo\_ SEQC.NEXTVAL,'Angie','2320');

COMMIT;

SELECT \* FROM Cargo\_Empleado;

--índice

CREATE INDEX Nom\_IdCargo\_idx ON Cargo\_Empleado(NombreC);

Select \* From Cargo\_Empleado Where NombreC = 'Angie';

SELECT \* FROM Cargo\_Empleado;

-- Alias

CREATE SYNONYM CargoX FOR Cargo\_Empleado;

**Pregunta 4:**

DROP TABLE NOTAS;

DROP TABLE ALUMNO;

DROP TABLE CURSOS;

DROP TABLE PROFESOR;

CREATE TABLE CURSO(

COD\_CURSO VARCHAR2(5 BYTE) NOT NULL,

DESCRIPCION VARCHAR2(30 BYTE) NOT NULL,

CREDITO NUMBER(\*, 0) NOT NULL,

ESTADO VARCHAR2(20 BYTE) NOT NULL,

CONSTRAINT PK\_CODIGO\_CURSO PRIMARY KEY (COD\_CURSO),

CONSTRAINT CHK\_DESCRIPCION\_CURSO CHECK (DESCRIPCION = NLS\_UPPER(DESCRIPCION)),

CONSTRAINT CHK\_CREDITO\_CURSO CHECK (CREDITO IN (2,3,4,6)),

CONSTRAINT CHK\_ESTADO\_CURSO CHECK (ESTADO IN ('ACTIVO','BLOQUEADO'))

);

CREATE TABLE PROFESOR(

COD\_PROF VARCHAR2(5 BYTE) NOT NULL,

NOMBRE VARCHAR2(50 BYTE) NOT NULL,

PROFESION VARCHAR2(50 BYTE) NOT NULL,

F\_NAC DATE NOT NULL,

DNI CHAR(8 BYTE) NOT NULL,

SEXO VARCHAR2(20 BYTE) NOT NULL,

EDAD NUMBER(\*, 0) NOT NULL,

ESTADO VARCHAR2(20 BYTE) NOT NULL,

CONSTRAINT PK\_CODIGO\_PROFESOR PRIMARY KEY (COD\_PROF),

CONSTRAINT CHK\_NOMBRE\_PROFESOR CHECK (NOMBRE = NLS\_UPPER(NOMBRE)),

CONSTRAINT CHK\_PROFESION\_PROFESOR CHECK (PROFESION = NLS\_UPPER(PROFESION)),

CONSTRAINT U\_DNI\_PROFESOR UNIQUE (DNI),

CONSTRAINT CHK\_SEXO\_PROFESOR CHECK (SEXO IN ('Masculino','Femenino')),

CONSTRAINT CHK\_EDAD\_PROFESOR CHECK (EDAD >= 18 AND EDAD <= 52),

CONSTRAINT CHK\_ESTADO\_PROFESOR CHECK (ESTADO IN ('ACTIVO','BLOQUEADO','SUSPENDIDO'))

);

CREATE TABLE ALUMNO(

COD\_ALU VARCHAR2(5 BYTE) NOT NULL,

NOMBRE VARCHAR2(50 BYTE) NOT NULL,

SEXO VARCHAR2(20 BYTE) NOT NULL,

GRADO NUMBER(\*, 0) NOT NULL,

F\_NACIMIENTO DATE NOT NULL,

EDAD NUMBER(\*, 0) NOT NULL,

ESTADO VARCHAR2(20 BYTE) NOT NULL,

CONSTRAINT PK\_CODIGO\_ALUMNO PRIMARY KEY (COD\_ALU),

CONSTRAINT CHK\_NOMBRE\_ALUMNO CHECK (NOMBRE = NLS\_UPPER(NOMBRE)),

CONSTRAINT CHK\_SEXO\_ALUMNO CHECK (SEXO IN ('MASCULINO','FEMENINO')),

CONSTRAINT CHK\_GRADO\_ALUMNO CHECK (GRADO IN (1,2,3,4,5,6)),

CONSTRAINT CHK\_EDAD\_ALUMNO CHECK (EDAD >= 10 AND EDAD <= 22),

CONSTRAINT CHK\_ESTADO\_ALUMNO CHECK(ESTADO IN ('ACTIVO','BLOQUEADO','SUSPENDIDO'))

);

CREATE TABLE NOTAS(

COD\_ALU VARCHAR2(5 BYTE) NOT NULL,

COD\_CURSO VARCHAR2(5 BYTE) NOT NULL,

COD\_PROF VARCHAR2(5 BYTE) NOT NULL,

NOTA1 NUMBER(\*, 0) NOT NULL,

NOTA2 NUMBER(\*, 0) NOT NULL,

NOTA3 NUMBER(\*, 0) NOT NULL,

NOTA4 NUMBER(\*, 0) NOT NULL,

FECHA DATE NOT NULL,

CONSTRAINT PK\_ALUMNO\_CURSO\_PROFESOR PRIMARY KEY (COD\_ALU, COD\_CURSO, COD\_PROF),

CONSTRAINT CHK\_NOTA1\_NOTAS CHECK (NOTA1 >= 0 AND NOTA1 <= 20),

CONSTRAINT CHK\_NOTA2\_NOTAS CHECK (NOTA2 >= 0 AND NOTA2 <= 20),

CONSTRAINT CHK\_NOTA3\_NOTAS CHECK (NOTA3 >= 0 AND NOTA3 <= 20),

CONSTRAINT CHL\_NOTA4\_NOTAS CHECK (NOTA4 >= 0 AND NOTA4 <= 20),

CONSTRAINT FK\_REL\_NOTAS\_ALUMNO FOREIGN KEY (COD\_ALU) REFERENCES ALUMNO(COD\_ALU),

CONSTRAINT FK\_REL\_NOTAS\_CURSO FOREIGN KEY (COD\_CURSO) REFERENCES CURSO(COD\_CURSO),

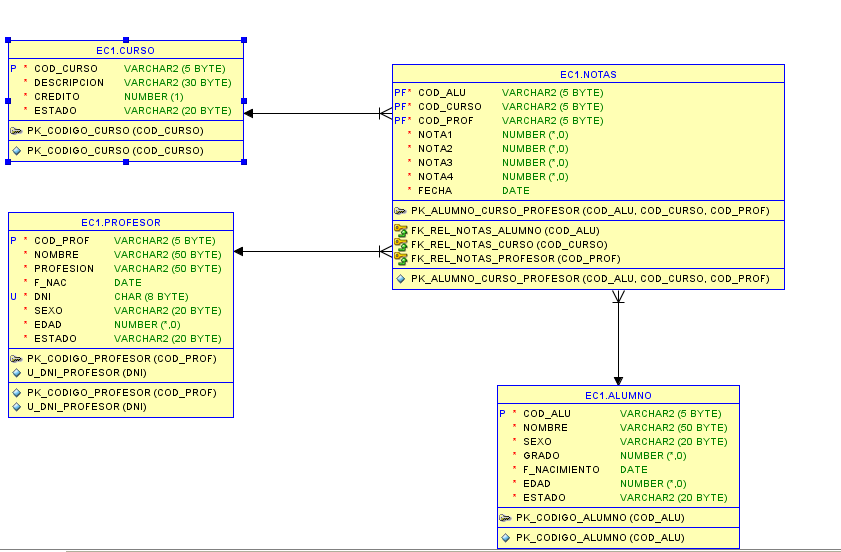
CONSTRAINT FK\_REL\_NOTAS\_PROFESOR FOREIGN KEY (COD\_PROF) REFERENCES PROFESOR(COD\_PROF)

);

SELECT \* FROM CURSO;

SELECT \* FROM PROFESOR;

SELECT \* FROM ALUMNO;

SELECT \* FROM NOTAS;

**Pregunta 5:**

-- PREGUNTA 5.A

SELECT AP\_TRA, AM\_TRA, NOM\_TRA, SUELDO,

(SUELDO\*0.09) AS ESSALUD, (SUELDO\*0.13) AS ONP,

SUELDO - (SUELDO\*0.13) AS SUELDO\_NETO

FROM TRABAJADOR

-- PREGUNTA 5.B

SELECT TITULOLIB, AUTORLIB, PRECIOLIB, PRECIOLIB\*0.18 AS IGV, PRECIOLIB+(PRECIOLIB\*0.18) AS PRECIO\_TOTAL

FROM LIBRO

-- PREGUNTA 5.C

SELECT SEXO, COUNT(Cod\_Alu) FROM ALUMNO GROUP BY SEXO

--PREGUNTA 5.D

SELECT AREALIB, SUM(PRECIOLIB)FROM LIBRO WHERE AREALIB IN ('ARTE','PROYECTOS','LPROG')

GROUP BY AREALIB

-- PREGUNTA 5.E

SELECT P.Nombre, C.Descripcion, Nota1, Nota2, Nota3, Nota4, (Nota1\*0.04)+(Nota2\*0.12)+(Nota3\*0.24)+(Nota4\*0.6) AS PROMEDIO

FROM PROFESOR P, CURSOS C, NOTAS N, ALUMNO A

WHERE P.Cod\_Prof = N.Cod\_Prof AND C.Cod\_Curso = N.Cod\_Curso AND A.Cod\_Alu = N.Cod\_Alu

AND A.Nombre LIKE '%PEREZ%'

ORDER BY P.Nombre, C.Descripcion

SELECT \* FROM ALUMNO

SELECT P.Nombre, C.DESCRIPCION, N.NOTA1, N.NOTA2, N.NOTA3, N.NOTA4,

(N.NOTA1\*0.04)+(N.NOTA2\*0.12)+(N.NOTA3\*0.24)+(N.NOTA4\*0.6) PROMEDIO

FROM NOTAS N NATURAL JOIN PROFESOR P

NATURAL JOIN CURSOS C

WHERE N.COD\_ALU = 'A0001'