**Manual de Prácticas de Laboratorio**

|  |  |  |
| --- | --- | --- |
| **Bases de datos II** | | |
| RESPONSABLE | Jesús Alejandro flores Hernández | |
| COLABORADORES | Academia de programación | |
| REVISADO POR: | Academia de Programación | |
| Materia | Bases de datos II | |
| Carreras | ISC, ICO | |
| APROBADO POR: | Academia de Programación | |
| FECHA:  16/06/2022 | FECHA DE REGISTRO: | TOTAL, DE PÁGINAS  Página |

Contenido

[Introducción. 5](#_Toc104298941)

[Conceptos. 5](#_Toc104298942)

[1.- Practica Bases de datos. 5](#_Toc104298943)

[Información y datos. 5](#_Toc104298944)

[Bases de datos. 6](#_Toc104298945)

[Manejadores de bases de datos. 7](#_Toc104298946)

[Esquema de la base de datos. 7](#_Toc104298947)

[Instancia de la base de datos. 7](#_Toc104298948)

[Objetivo de la práctica. 1. 7](#_Toc104298949)

[materiales de la práctica. 1. 7](#_Toc104298950)

[desarrollo de la práctica. 1. 7](#_Toc104298951)

[entregables de la práctica. 1. 8](#_Toc104298952)

[2.- Práctica Modelo Entidad Relación. 8](#_Toc104298953)

[introducción. 8](#_Toc104298954)

[Simbología 9](#_Toc104298955)

[Objetivo de la práctica 2 10](#_Toc104298956)

[Materiales de la práctica 2 10](#_Toc104298957)

[Desarrollo de la práctica 2 10](#_Toc104298958)

[Entregables de la práctica 2 10](#_Toc104298959)

[3.- Práctica Relaciones entre entidades modelo ER. 11](#_Toc104298960)

[Relaciones 11](#_Toc104298961)

[Relación 1:1 11](#_Toc104298962)

[Relación 1: N 11](#_Toc104298963)

[Relación muchos a muchos 11](#_Toc104298964)

[Objetivo de la práctica 3 12](#_Toc104298965)

[Materiales de la práctica 3 12](#_Toc104298966)

[Desarrollo de la práctica 3 12](#_Toc104298967)

[Entregables de la práctica 3 13](#_Toc104298968)

[4.- Atributos en las relaciones. 13](#_Toc104298969)

[Objetivo de la práctica 4 13](#_Toc104298970)

[Materiales de la práctica 4 13](#_Toc104298971)

[Desarrollo de la práctica 4 14](#_Toc104298972)

[Entregables de la práctica 4 14](#_Toc104298973)

[5.- del Modelo ER a la base de datos. 14](#_Toc104298974)

[construir tablas 14](#_Toc104298975)

[llaves de las tablas 15](#_Toc104298976)

[cardinalidad de la relacion 17](#_Toc104298977)

[Objetivo de la práctica 5 19](#_Toc104298978)

[Materiales de la práctica 5 19](#_Toc104298979)

[Desarrollo de la práctica 5 19](#_Toc104298980)

[Entregables de la práctica 5 19](#_Toc104298981)

[6.- Modelo relacional. 19](#_Toc104298982)

[6.1.- Relaciones y dominios. 20](#_Toc104298983)

[6.2.- reglas de integridad relacional. 20](#_Toc104298984)

[6.2.1.- Calves primarias. 21](#_Toc104298985)

[6.2.2.- Regla de integridad de las entidades. 21](#_Toc104298986)

[6.2.3.- Claves foráneas. 22](#_Toc104298987)

[6.3.- Álgebra relacional. 22](#_Toc104298988)

[6.3.1.- Operaciones con conjuntos. 22](#_Toc104298989)

[6.3.2.- Proyección, SELECCIÓN, producto cartesiano, diferencia. 23](#_Toc104298990)

[6.4.- Cálculo relacional. 24](#_Toc104298991)

[1.- La operacionde seleccion. 24](#_Toc104298992)

[2 La operación de proyección 25](#_Toc104298993)

[3 La operación de interseccion 26](#_Toc104298994)

[Objetivo de la práctica 6.1 26](#_Toc104298995)

[Materiales de la práctica 6.1 26](#_Toc104298996)

[Desarrollo de la práctica 6.1 26](#_Toc104298997)

[Entregables de la práctica 6 27](#_Toc104298998)

[6.5.- Uso de un DBMS en linea de comandos. 27](#_Toc104298999)

[1.- Instalar XAMP 27](#_Toc104299000)

[Conéctese al DBMS (database manager system) con el comando: 28](#_Toc104299001)

[Comandos 29](#_Toc104299002)

[Objetivo de la práctica 6.2. 31](#_Toc104299003)

[Materiales de la práctica 6.2 31](#_Toc104299004)

[Desarrollo de la práctica 6.2 31](#_Toc104299005)

[7.- Diseño de bases de datos modelo relacional. 32](#_Toc104299006)

[1.- El enfoque intuitivo. 32](#_Toc104299007)

[Objetivo de la práctica 7.1. 34](#_Toc104299008)

[Materiales de la práctica 7.1. 34](#_Toc104299009)

[Desarrollo de la práctica 7.1. 34](#_Toc104299010)

[Entregable de la práctica 7.1. 34](#_Toc104299011)

[2.- Normalización. 35](#_Toc104299012)

[1.- Primera forma normal. 35](#_Toc104299013)

[2.- Segunda forma normal. 35](#_Toc104299014)

[3.- forma normal de Boyce codd. 36](#_Toc104299015)

[Objetivo de la práctica. 7.2. 37](#_Toc104299016)

[materiales de la práctica 7.2. 37](#_Toc104299017)

[desarrollo de la práctica. 7.2. 37](#_Toc104299018)

[entregables de la práctica. 7.2. 37](#_Toc104299019)

[8.- Sistemas manejadores de bases de datos (dbms). 37](#_Toc104299020)

[1.- Data base manager system. 38](#_Toc104299021)

[2.- Creación de la base de datos. 38](#_Toc104299022)

[3.- Catalogo del sistema. 38](#_Toc104299023)

[4.- Vistas. 38](#_Toc104299024)

[5.- Sql. 38](#_Toc104299025)

[6.- Las doce reglas DE. CODD. 38](#_Toc104299026)

[7.- Recuperación y concurrencia. 40](#_Toc104299027)

[8.- Administración. 40](#_Toc104299028)

[9.- Lenguaje de Consultas. 40](#_Toc104299029)

[1.- Sentencia Create Table. 40](#_Toc104299030)

[3.- Sentencia Select. 41](#_Toc104299031)

[4.- sentencia insert. 42](#_Toc104299032)

[5.- sentencia update. 43](#_Toc104299033)

[6.- Sentencia Like. 43](#_Toc104299034)

[Objetivo de la práctica. 9. 43](#_Toc104299035)

[Materiales de la práctica. 9. 44](#_Toc104299036)

[Desarrollo de la práctica. 7.2. 44](#_Toc104299037)

[Entregable de la práctica. 7.2. 44](#_Toc104299038)

[10.- Administracion. 44](#_Toc104299039)

[1. Metadatos 44](#_Toc104299040)

[2 Seguridad en bases de datos. 48](#_Toc104299041)

[3 Seguridad en bases de datos. 48](#_Toc104299042)

# Introducción.

Bienvenidos al manual de prácticas de la materia de Bases de Datos del MI Jesús Alejandro Flores Hernández daremos una breve introducción a las bases de datos.

El procesamiento de la información requiere de almacenes de datos confiables y organizados de tal manera que nos provean información consistente, en un sistema de cómputo no nos interesa que la información sea correcta, pues esto depende del capturista, si almaceno un dato X sea incorrecto o no cada que consulte los bancos de datos debemos darle el mismo valor X, eso es consistencia sin embargo un objetivo tan sencillo requiere de cierto grado de complejidad, esta complejidad es manejada por la organización que le hayamos dato a nuestra base de datos.

## Conceptos.

Base de datos es un conjunto de tablas relacionadas entre sí.

Tabla es un conjunto de valores que responde a una relación, por esta razón a una tabla también se le llama relación.

Atributos son los campos de una tabla.

Llave es un atributo de la tabla que identifica de manera única a los renglones de una tabla.

Relación es una tabla.

Tupla es un renglón de la tabla.

DBMS o SMBD sistema manejador de base de datos (data base manager system) es un conjunto de programas que permite manejar una base de datos. Por ejemplo, MySQl, SQL SERVER, ORACLE.

# 1.- Practica Bases de datos.

## Información y datos.

La información nos describe conocimientos, procesos, hechos y normalmente el humano la representa en lenguaje coloquial, que puede tener más o menos palabras según su redactor, en el ámbito del almacenamiento en equipos de cómputo, debemos buscar una representación de la información, que sea clara y concisa, por ejemplo podemos decir: “El señor Juan Pérez Gutiérrez tiene 39 años cumplidos y su ocupación es profesor”, esto podría representarse (para leer: como una tabla con 4 columnas donde en la primera columna la titulamos nombre, la segunda sexo, la tercera edad y la cuarta ocupación y en cada renglón o fila ponemos los datos de cada persona con su nombre, sexo, edad y ocupación en las respectivas columnas) así:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Sexo | Edad | Ocupación |
| Juan Pérez Gutiérrez | M | 39 | Profesor |

Tabla 1.1.1

Es la misma información, pero en la tabla solo se representa lo indispensable, además de que la información está separada en datos como Nombre, Sexo, etc... Si tomamos el sexo por sí solo no nos dice mucho, solo que hablamos de una persona de sexo masculino en este caso, pero en su contexto (o sea en su renglón) nos da información sobre una persona, cuando a un dato suelto lo analizamos en su contexto tenemos información, sin el contexto es solo un dato, en nuestro ejemplo cada columna es un campo y cada renglón es un registro, aquí vemos 4 campos (nombre, sexo, edad y ocupación) por cada registro.

Un Campo es una unidad indivisible para un contexto dado, es un dato, pensemos en el como un Token o unidad mínima consumible por la aplicación, un campo no tiene un tamaño fijo, podemos decir que es un conjunto de símbolos que pierden sentido al quitar uno de ellos. Por ejemplo en el nombre de una persona, al quitarle una letra a ese elemento, este pierde sentido, un conjunto de campos define un registro y todos los campos del registro guardan información sobre un mismo elemento, los elementos o registros de una tabla deben ser único e irrepetibles en la tabla 1.1.1 se almacenan registros de personas con 4 campos que describen a la persona, estos campos pueden ser suficientes en un contexto como el seno de una familia donde difícilmente 2 personas tendrán los mismos datos, es decir puede que tengan la misma edad (si son gemelos) pero no el mismo nombre, esto es importante porque dijimos que los registros deben ser únicos e irrepetibles es decir deben ser diferentes al menos en un campo, sin embargo en un grupo de empleados de una empresa (PEMEX por ejemplo) no bastan estos 4 campos, pues puede darse el caso (raro si se quiere pero se da) de 2 personas diferentes con el mismo nombre, el mismo sexo, la misma edad y la misma ocupación. Para estos casos los campos de nuestro ejemplo ya no son suficientes, debemos incluir al menos un campo más que haga a los registros irrepetibles, por ejemplo, el RFC de la persona o el número de empleado.

Un registro es un conjunto de campos con relación entre sí, lo podemos pensar como el conjunto de atributos pertinentes a una entidad, por ejemplo, si hablamos de la entidad alumno, sus atributos son (matricula, nombre, semestre, etc.).

La información para una entidad, se puede representar como una tabla en la cual cada columna representa un campo y cada fila (o renglón) representa un registro que describe a una entidad en particular.

## Bases de datos.

Una base de datos es una colección de archivos interrelacionados y estructurados que nos ayuda a garantizar la consistencia de la información.

En una empresa tendremos un conjunto de entidades descritas cada una por una tabla y todas esas tablas estarán interrelacionadas de alguna forma, por ejemplo, la tabla empleado estará relacionado con la tabla nomina porque en la nómina aparece el número de empleado, la tabla pedidos estará relacionada con la tabla partes por que en la tabla pedidos tenemos la información de la parte que se ha pedido, en estas notas trabajaremos sobre el modelo relacional de base de datos y nuestra tarea será construir un conjunto de tablas interrelacionadas que describan la información de una empresa.

## Manejadores de bases de datos.

Un manejador de base de datos (DBMS data base manager system) es un conjunto de programas que permiten administrar una base de datos. Entre los DBMS más usados están: Oracle, Infomix, SqlServer, MySql. Y existen un conjunto de mini manejadores para aplicaciones ligeras tales como Acces, SQLite, JabaDB.

## Esquema de la base de datos.

Es el diseño de la base de datos, es el modelo con el cual representamos la información y esto incluye un conjunto de relaciones y otros elementos como reglas de integridad referencial.

## Instancia de la base de datos.

En el conjunto de datos almacenados según el esquema previamente definido.

## Objetivo de la práctica. 1.

Identificar los tipos de DBMS relacionales y no relacionales, mostrar las características de los 4 principales manejadores de base de datos relacionales y 2 no relacionales. Para el ámbito de la FCI identifique las tablas necesarias para llevar el control de los datos de alumno, cursos y carreras de la institución.

## materiales de la práctica. 1.

* Computadora
* Internet
* Suite de programas de oficina

## desarrollo de la práctica. 1.

1. Investigar en internet los principales manejadores e base de datos, identificando:
   1. Libre o de licencia
   2. Capacidades de almacenamiento
   3. Requerimientos de hardware
   4. Frecuencia de uso (que porcentaje del mercado tiene)
2. Hacer una tabla comparativa del resultado de su investigación
3. Repetir el paso 1 y 2 para los DBMS no relacionales
4. Agregue a su trabajo las referencias de la investigación.
5. Describa mediante tablas las entidades alumno, profesor, materia y carrera.

## entregables de la práctica. 1.

* Archivo pdf con el resultado de su investigación.
* Archivo pdf con el resultado de sus tablas.
* Archivo .doc con el resultado de su invetigación

# 2.- Práctica Modelo Entidad Relación.

## introducción.

El modelo entidad relación es una herramienta que permite representar de manera simplificada los componentes que participan en un proceso de negocio y el modo en el que estos se relacionan entre sí.

El modelo entidad relación tiene tres elementos principales:

Entidades: El modelo contará con una entidad por cada uno de los componentes del proceso de negocio. Así, en un negocio de venta de suscripciones a revistas, podemos tener entidades “Cliente”, “Dirección”, “Factura”, “Producto”, o “Incidencias”, entre otras.

Atributos: Los atributos, componente fundamental de cada modelo entidad-relación, nos permiten describir las propiedades que tiene cada entidad. “Nombre”, “Primer Apellido”, “Segundo Apellido”,” Fecha de nacimiento”, “Género” o “Segmento de valor” serán atributos de la entidad “Cliente”.

Relaciones: Con las relaciones se establecen vínculos entre parejas de entidades. Cada “Cliente” tendrá una “Dirección” de envío en la que recibirá la suscripción, podrá estar suscrito a uno o varios “Productos”, y recibirá una “Factura” con la periodicidad acordada.

El diagrama entidad relación es la expresión gráfica del modelo entidad relación. En él las entidades se representan utilizando rectángulos, los atributos por medio de círculos o elipses y las relaciones como líneas que conectan las entidades que tienen algún tipo de vínculo. También es muy común el formato de diagrama en el que los atributos de una entidad aparecen listados en filas dentro del rectángulo que representa a esa entidad.

Además, es común que, en el modelo entidad-relación, los conectores que indican que dos entidades A y B están relacionadas entre sí tengan una apariencia gráfica diferente dependiendo del tipo de relación que exista entre ellas.

Los tipos de relaciones posibles entre dos entidades en un modelo entidad relación son:

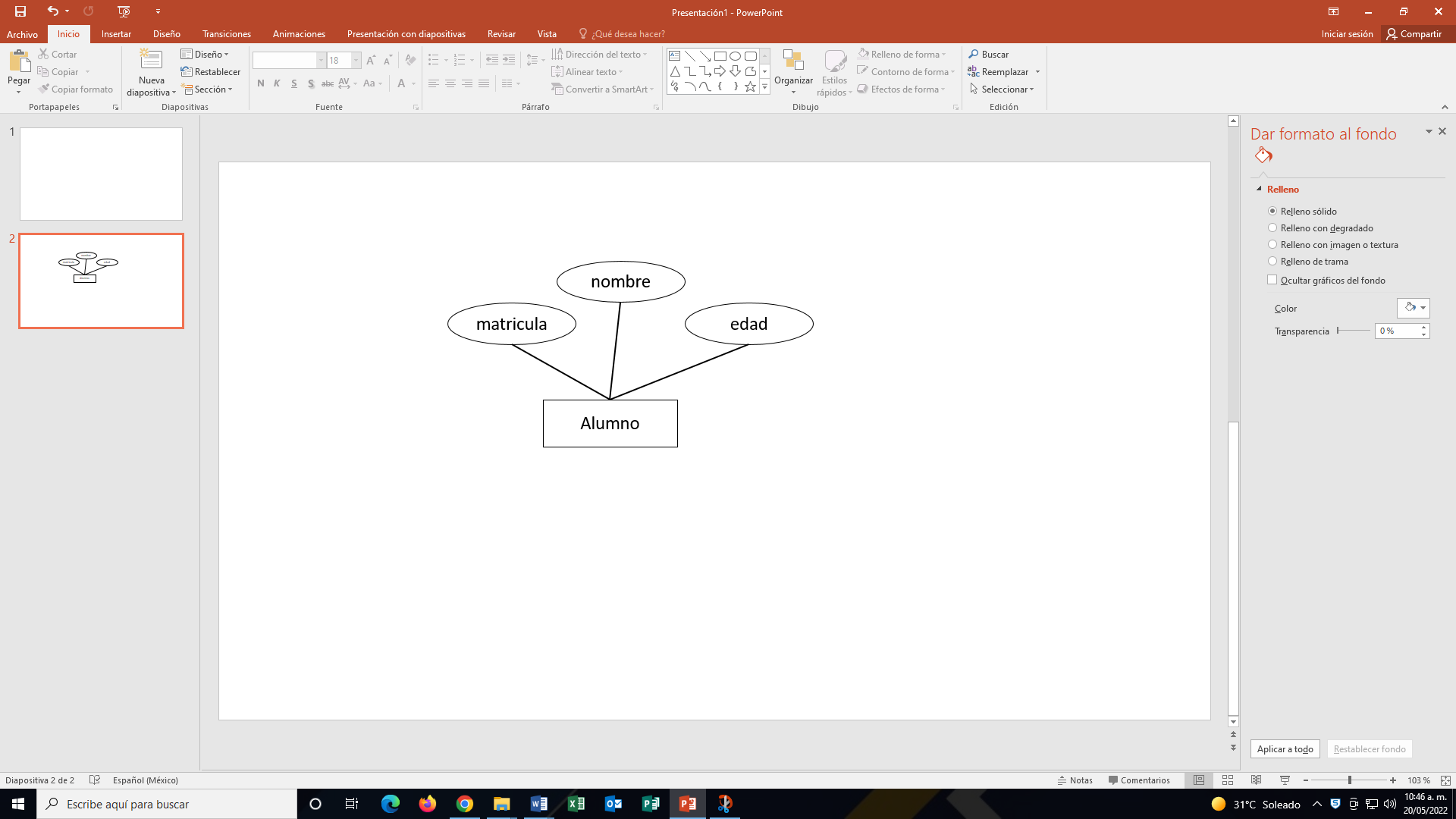
Relación uno a uno: Un “individuo” de la entidad A solamente puede estar relacionado con un “individuo” de la entidad B, y ese “individuo” de la entidad B no puede estar relacionado con otros “individuos” de la entidad A. Por ejemplo, cada miembro de la entidad País se relaciona únicamente con un miembro de la entidad “Ciudad capital de un país”. Cada país puede tener una única capital y cada ciudad capital puede serlo únicamente de un país.

Relación uno a varios: Un “individuo” de la entidad A puede estar relacionado con uno o varios “individuos” de la entidad B, y esos “individuos” de la entidad B no pueden estar relacionados con otros “individuos” de la entidad A. Por ejemplo, cada miembro de la entidad “Padre” puede estar relacionado con uno o varios miembros de la entidad “Hijo”, y cada miembro de la entidad “Hijo” solamente puede tener vínculo con un miembro de la entidad “Padre”.

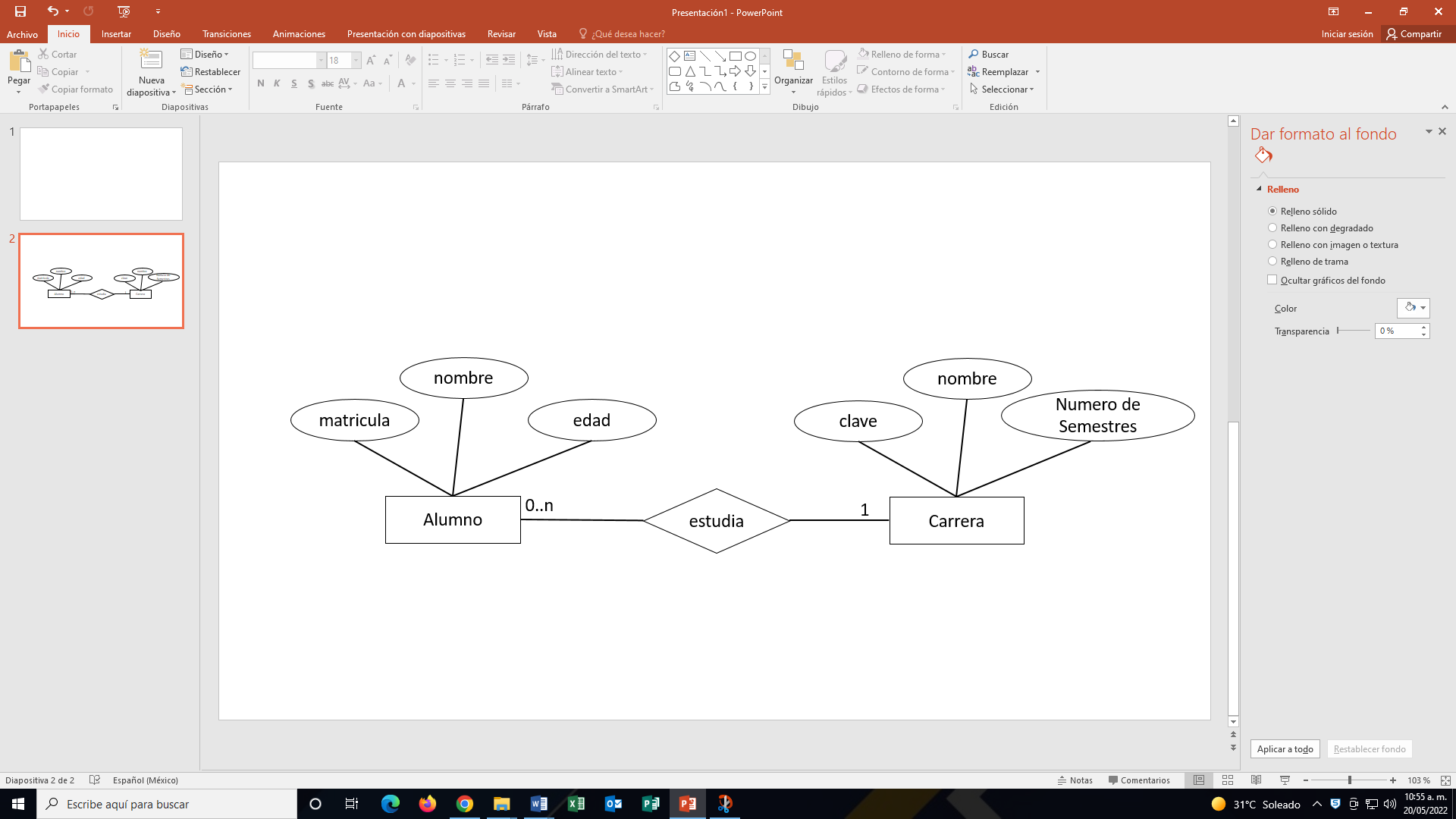
Relación varios a varios: Cada “individuo” de la entidad A puede estar relacionado con uno o varios “individuos” de la entidad B, y cada “individuo” de la entidad B puede estar relacionado con varios “individuos” de la entidad A. Por ejemplo, cada miembro de la entidad “Cliente” puede estar relacionado con uno o varios miembros de la entidad “Producto”, y cada miembro de la entidad “Producto” puede tener vínculo con varios miembros de la entidad “Cliente”.

## Simbología

Una entidad está compuesta por un conjunto de atributos que la describen, esos atributos dependen del ámbito en el que se describe la entidad por ejemplo en una escuela un alumno tendría atributos: matricula, nombre y sexo. Los símbolos que describen la entidad con sus atributos son:



Suponga que se tiene la entidad carrera con atributos clave, nombre y número de semestres y sabemos que un alumno tiene una relación con una carrera y es una relación de estudio, esto se representa así:



Este diagrama E-R dice que en una carreta estudian de 0 a n alumnos.

## Objetivo de la práctica 2

Diseñar el diagrama de relaciones de un problema de bases de datos.

## Materiales de la práctica 2

* Computadora
* Internet
* Suite de programas de oficina

## Desarrollo de la práctica 2

Practica 2.1

Estudie la narrativa de proveedores y partes proporcionada por el profesor en el curso y que también encontrará en los materiales de trabajo, a continuación, realice:

1. Identifique entidades y atributos, las entidades y atributos puede localizarlos subrayando adjetivos y sustantivos en su narrativa, tome como ejemplo el ejercicio mostrado en la sección “7.1 diseño intuitivo” de este mismo material para comprender mejor este punto.
2. Represente los diagramas de entidades y atributos.
3. Agregue las relaciones que su estudio indique.
4. Con esa información construya un diagrama ER.

Practica 2.2

1. Investigue el funcionamiento de los procesos de gestión académica en la FCI.
2. Escriba una narrativa que describa su problema.
3. Identifique entidades y atributos, las entidades y atributos puede localizarlos subrayando adjetivos y sustantivos en su narrativa, tomo como ejemplo el ejercicio mostrado en la sección “7.1 diseño intuitivo” de este mismo material para comprender mejor este punto.
4. Represente los diagramas de entidades y atributos.
5. Agregue las relaciones que su estudio indique.
6. Con esa información construya un diagrama ER.

## Entregables de la práctica 2

1. Archivo pdf con el diagrama ER
2. Archivo fuente con el diagrama ER

# 3.- Práctica Relaciones entre entidades modelo ER.

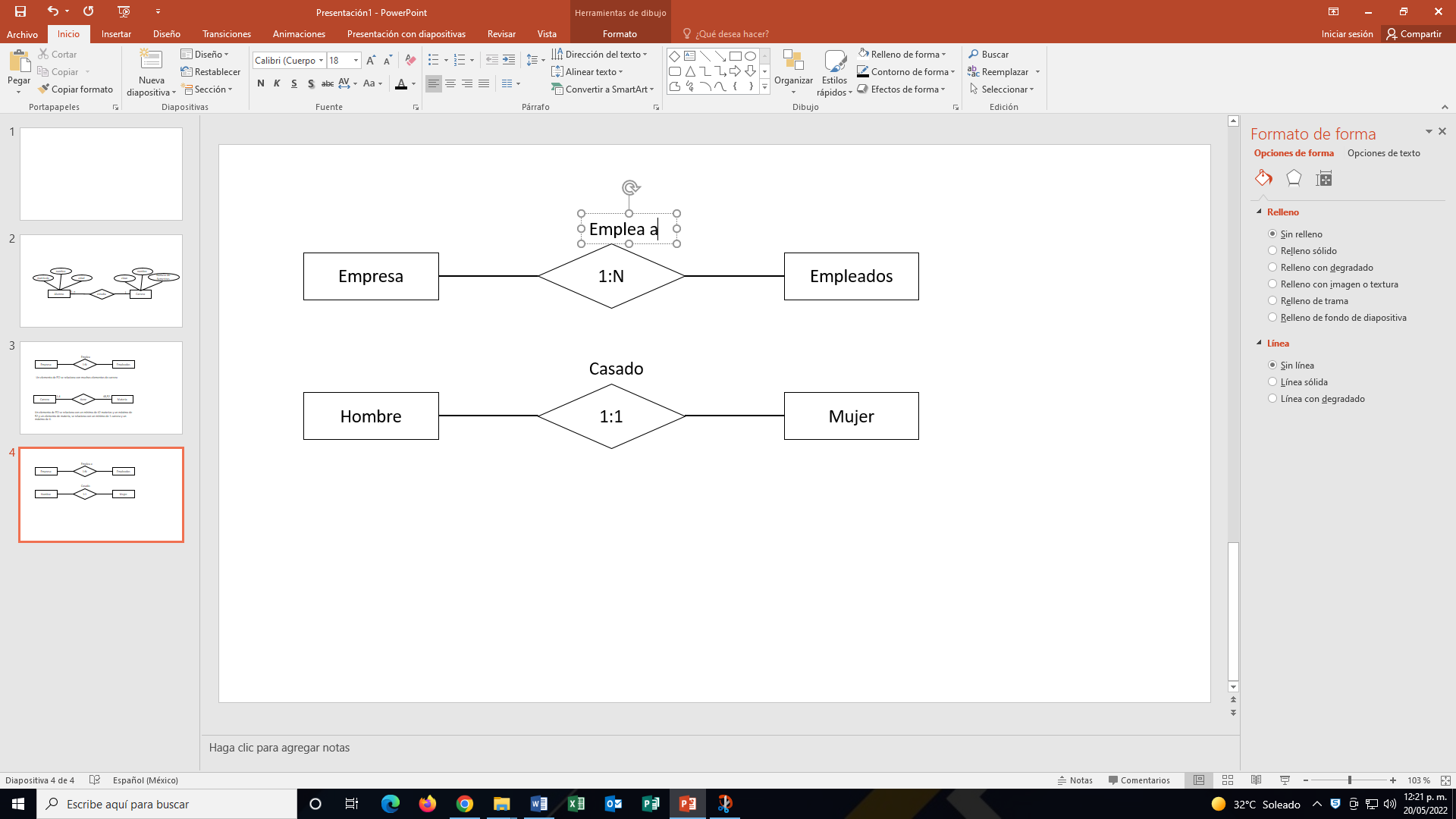
## Relaciones

Las entidades pueden tener las siguientes relaciones:

* Uno a uno (1:1).
* Uno a muchos (1:N).
* Muchos a muchos (N:M).

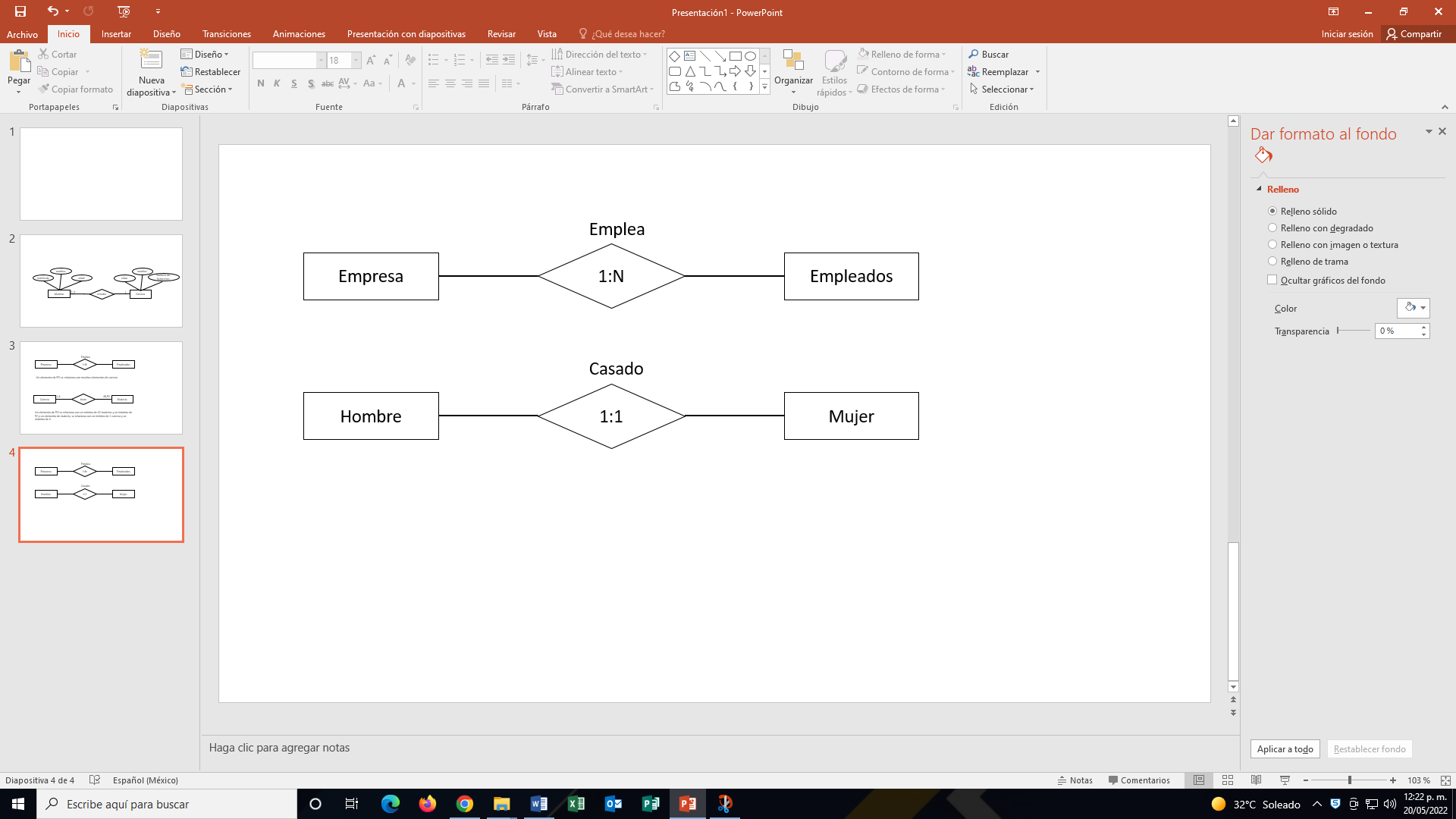
## Relación 1:1

Ejemplo: un hombre se relaciona con una mujer en la relación llamada “casado”:



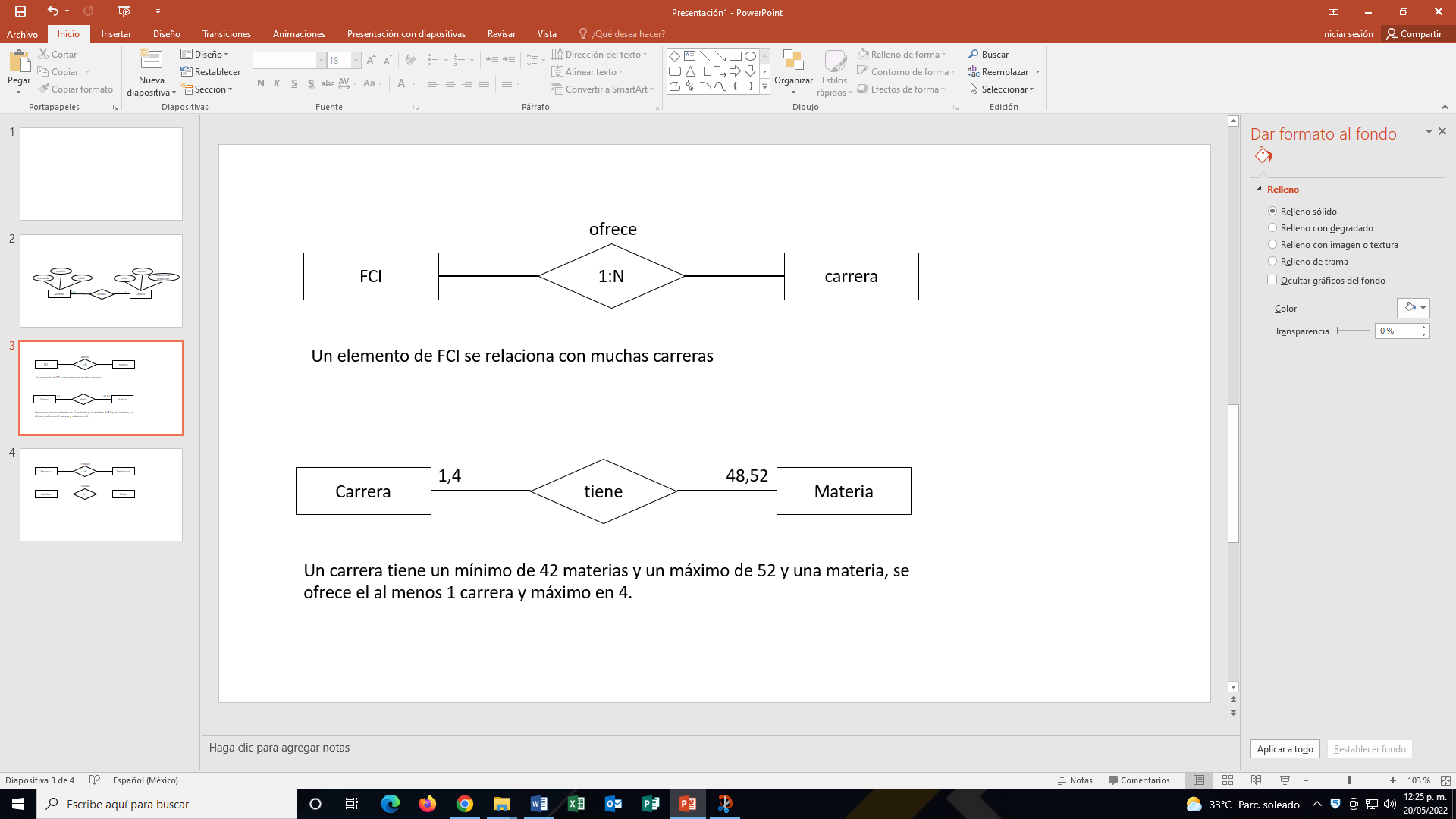
## Relación 1: N

Una empresa emplea a muchas personas en la relación “emplea”:



## Relación muchos a muchos

Ejemplos: Suponga las siguientes premisas: En la FCI se pueden ofrecer de 1 a N carreras según las necesidades de la población. Y en la FCI una carrera puede tener de 48 a 52 materias, y una materia se imparte en 1 o 4 carreras, esto último por que recuerde que hay materias especializantes que solo se llevan en 1 carrera y hay materias que se comparten entre carreras, estas premisas se diagraman así:



## Objetivo de la práctica 3

Diseñar el diagrama de relaciones de un problema de bases de datos.

## Materiales de la práctica 3

* Computadora
* Internet
* Suite de programas de oficina

## Desarrollo de la práctica 3

Practica 3.1

1. Reevalúe el funcionamiento de los procesos de la narrativa provPar.
2. Rediseñe los diagramas de entidades y atributos.
3. Agregue las relaciones que su estudio indique.
4. Con esa información construya un diagrama ER ahora mostrando las asociaciones correctas.

Practica 3.1

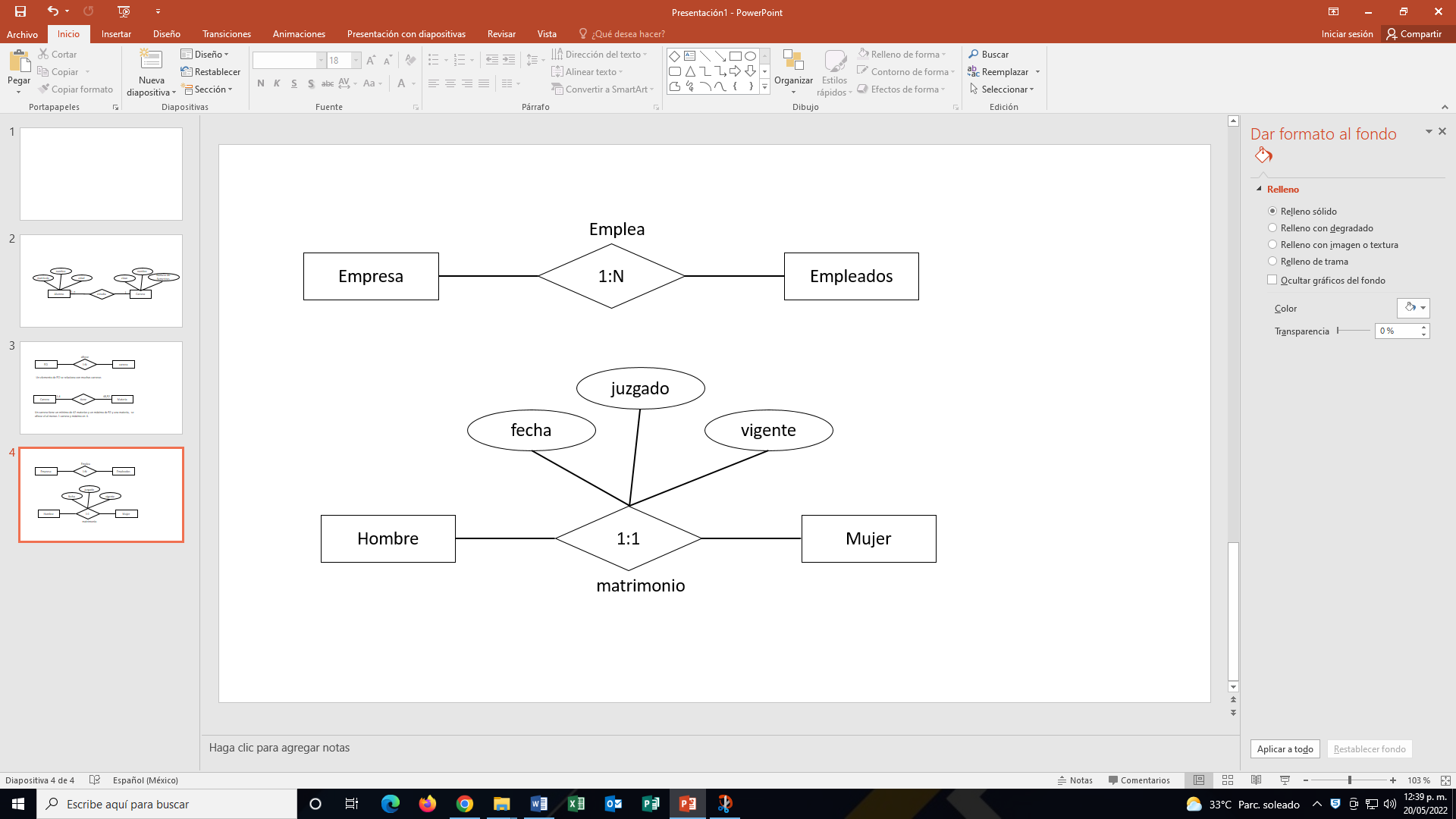
1. Reevalúe el funcionamiento de los procesos de gestión académica en la FCI que trabajo en la práctica 2.
2. Rediseñe los diagramas de entidades y atributos.
3. Agregue las relaciones que su estudio indique.
4. Con esa información construya un diagrama ER ahora mostrando las asociaciones correctas.

## Entregables de la práctica 3

1. Archivo pdf con el diagrama ER
2. Archivo fuente con el diagrama ER

# 4.- Atributos en las relaciones.

En ocasiones se requiere tener información en las relaciones entre entidades, por ejemplo, en la relación matrimonio que veíamos antes, tal vez se requieran datos como fecha del matrimonio, numero de juzgado y saber si está vigente o ya hubo divorcio. Esto se representa así:



Por ejemplo: en el control escolar sobre el que está trabajando puede tener la relación alumno materia, la relación se llamaría “curso” indicando que un alumno curso una materia y seguramente es necesario tener atributos del curso que tomo el alumno donde se indique el profesor, el período y la calificación obtenida, con esto en mente realice:

## Objetivo de la práctica 4

Diseñar el diagrama de relaciones de un problema de bases de datos.

## Materiales de la práctica 4

* Computadora
* Internet
* Suite de programas de oficina

## Desarrollo de la práctica 4

Práctica 4.1

1. Reevalúe el funcionamiento de los procesos de la narrativa ptovPar.
2. Rediseñe los diagramas de entidades y atributos.
3. Agregue los atributos a las relaciones donde sea necesario.
4. Con esa información construya un diagrama ER ahora mostrando los nuevos datos.

Práctica 4.2

1. Reevalúe el funcionamiento de los procesos de gestión académica en la FCI.
2. Rediseñe los diagramas de entidades y atributos.
3. Agregue los atributos a las relaciones donde sea necesario.
4. Con esa información construya un diagrama ER ahora mostrando los nuevos datos.

## Entregables de la práctica 4

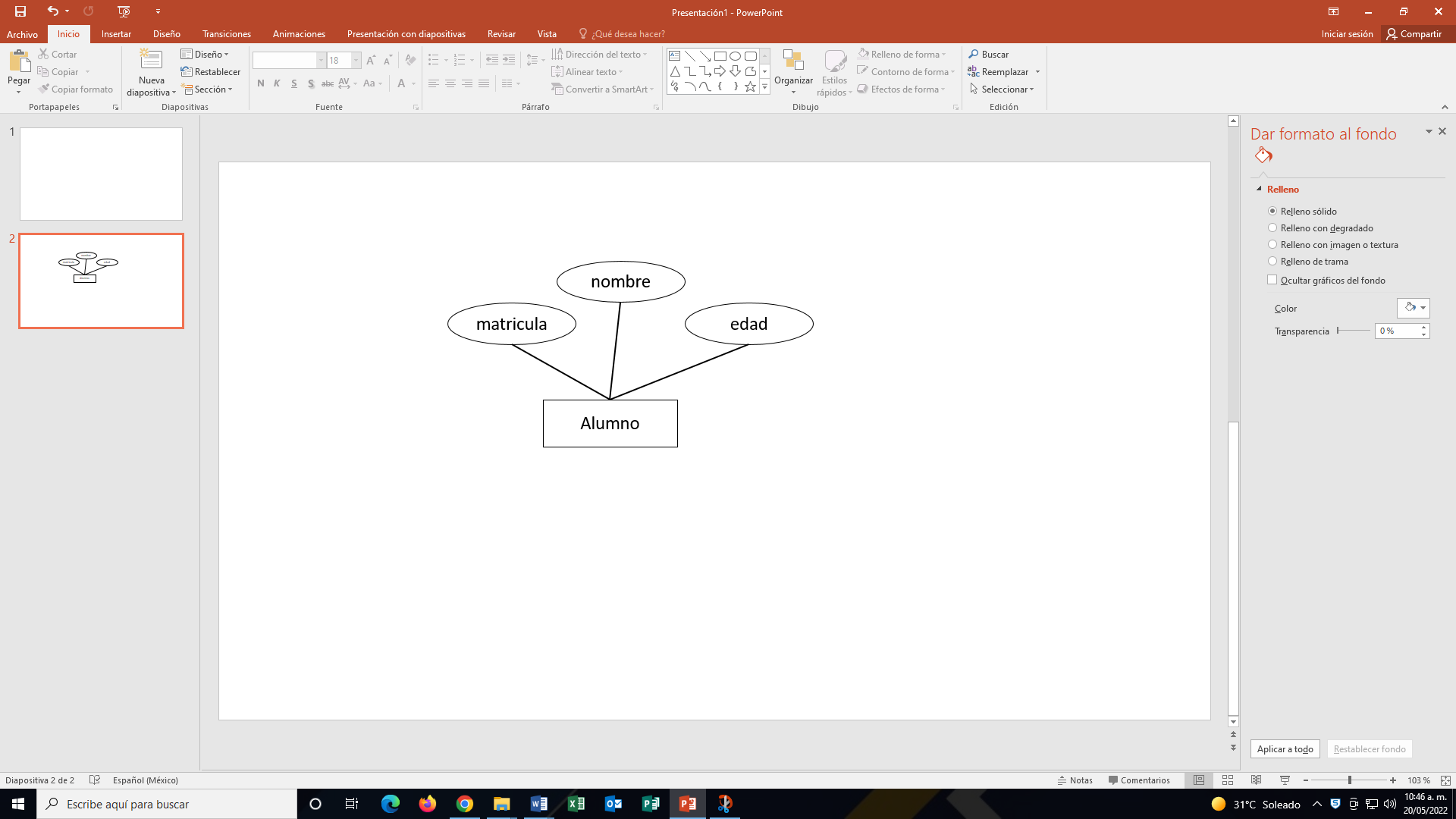
1. Archivo pdf con el diagrama ER
2. Archivo fuente con el diagrama ER

# 5.- del Modelo ER a la base de datos.

Para construir el esquema de base de datos donde necesitamos tener tablas debemos saber lo siguiente:

## construir tablas

**Primero**: los datos se representan en forma de tabla con las columnas representando los atributos de la entidad a los que llamaremos también campos de la tabla y los renglones representando las instancias de información de la tabla. Por ejemplo: la relación:



Se representa, así como una tabla:

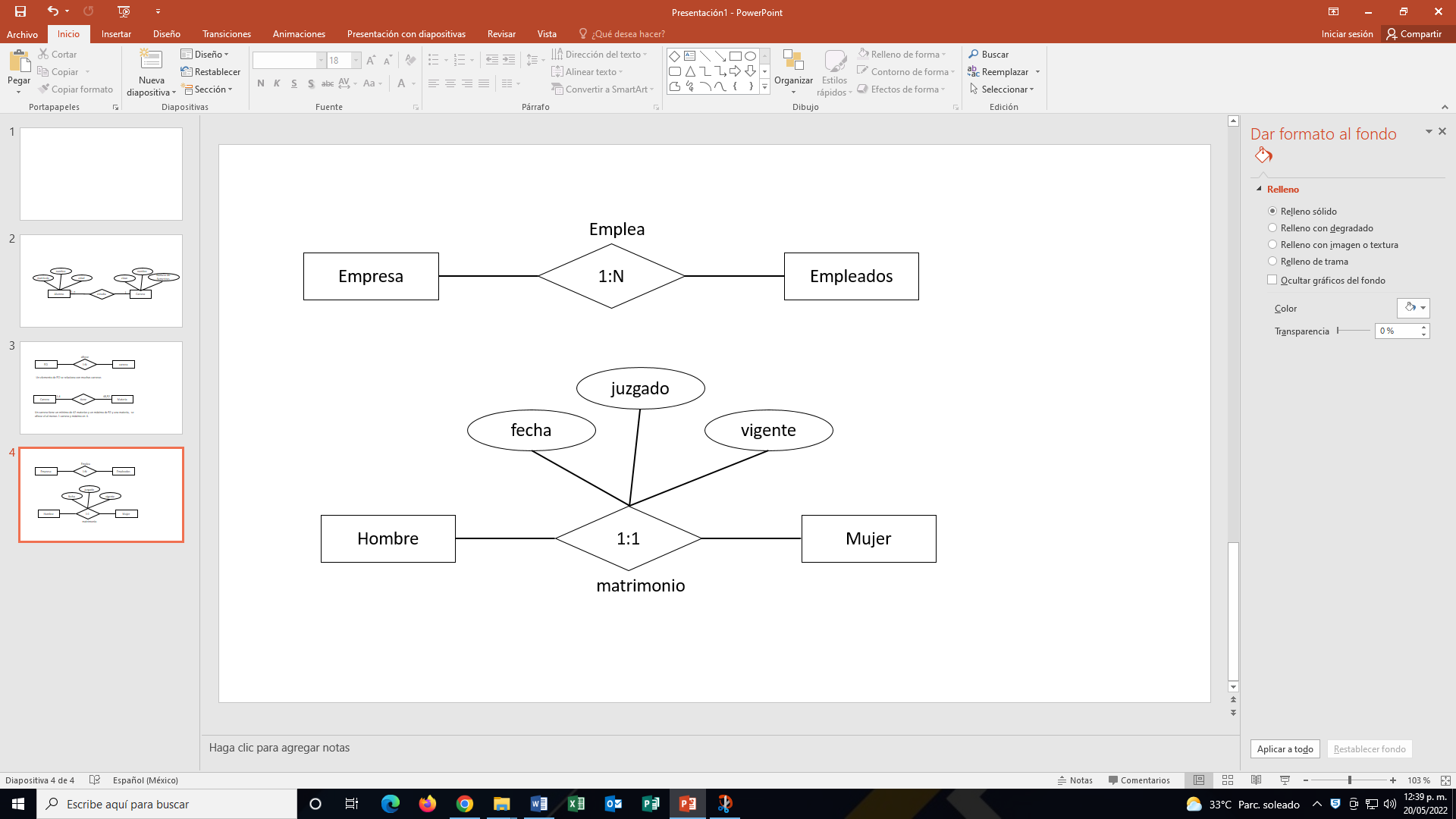
Tabla alumno:

|  |  |  |
| --- | --- | --- |
| Matricula | Nombre | edad |
| 202354 | Juan Perez | 22 |
| 217645 | Ana Uc | 19 |
| 221865 | Reina Puch | 22 |

## llaves de las tablas

**Segundo**: cada tabla debe tener una llave primaria, una llave es un campo o conjunto de campos que son irrepetibles en la tabla, por ejemplo: en esta tabla como ve las edades se pueden repetir, por tanto: el campo edad no puede ser llave, los nombres se pueden repetir, pues hay personas que se llaman igual, por lo que el campo nombre tampoco puede ser llave, y el campo matrícula es único para cada alumno, este si puede ser una llave.

Ahora si las relaciones tienen atributos como en:



Se representan las tablas hombre y mujer con sus atributos y la relación también se convierte en una tabla así:

Tabla hombre

|  |  |  |
| --- | --- | --- |
| rfc | Nombre | edad |
| Jupe546721 | Juan Perez | 22 |
| Pc765432 | Pedro Uc | 19 |
| Lp896534 | Luis Puch | 22 |

Tabla mujer

|  |  |  |
| --- | --- | --- |
| rfc | Nombre | edad |
| Bp123465 | Berta Perez | 23 |
| Au785423 | Ana Uc | 21 |
| Rp097645 | Reina Puch | 22 |

Se identifica la llave primaria de cada tabla, en este caso serán los campos rfc en ambas tablas. A continuación la relación se convierte en una tabla donde los primeros 2 campos corresponden a las llaves primarias de cada entidad que une esta relación en este caso la llave primaria d ela tabla hombre y la llave primaria de la tbala mujer, a continuación se agregan los campos de la relación.

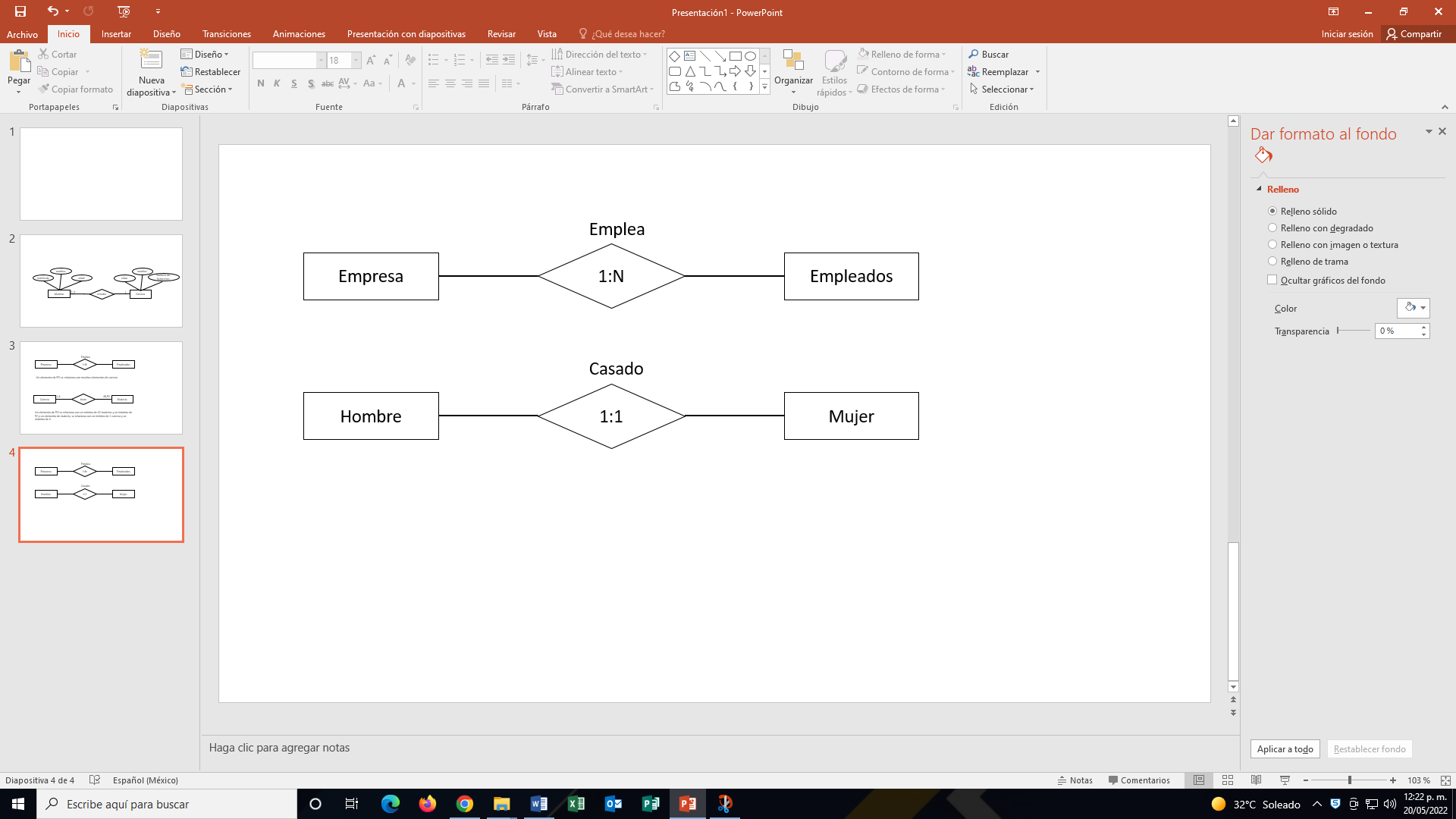
Tabla matrimonio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Llp\_hombre | Llp\_mujer | fecha | juzgado | vigente |
| Jupe546721 | Au785423 | 22/01/99 | 1 | si |
| Lp896534 | Bp123465 | 19/02/21 | 2 | no |

El segundo renglón de la tabla matrimonio dice que juan perez se casó con ana uc el 19 de febrero del 2021 y actualmente están divorciados.

Si la relación no tiene atributos, las tablas se construyen así:

Tome como ejemplo la relación:



Se construyen las tablas para ambas entidades:

Tabla empresa

|  |  |  |
| --- | --- | --- |
| rfc | nombre | dirección |
| Alu1234 | Aluminios del Carmen | 22 |
| Pc7982 | Panuchos chito | 19 |

Tabla empleado

|  |  |  |
| --- | --- | --- |
| rfc | nombre | especialidad |
| Bp123465 | Berta Perez | soldador |
| Au785423 | Ana Uc | cocinera |

## cardinalidad de la relacion

A continuación: se revisa la cordialidad de la relación:

Si es 1: N se agrega en la tabla “1” un campo para almacenar la llave primaria de la tabla “n”.

Si es 1: 1 se toma la primera tabla y se le agrega un campo para la llave primaria de la segunda.

Si es N:M se crea una nueva tabla con 2 campos donde cada campo de la nueva tabla corresponde a las llaves primarias de las tablas que relaciona.

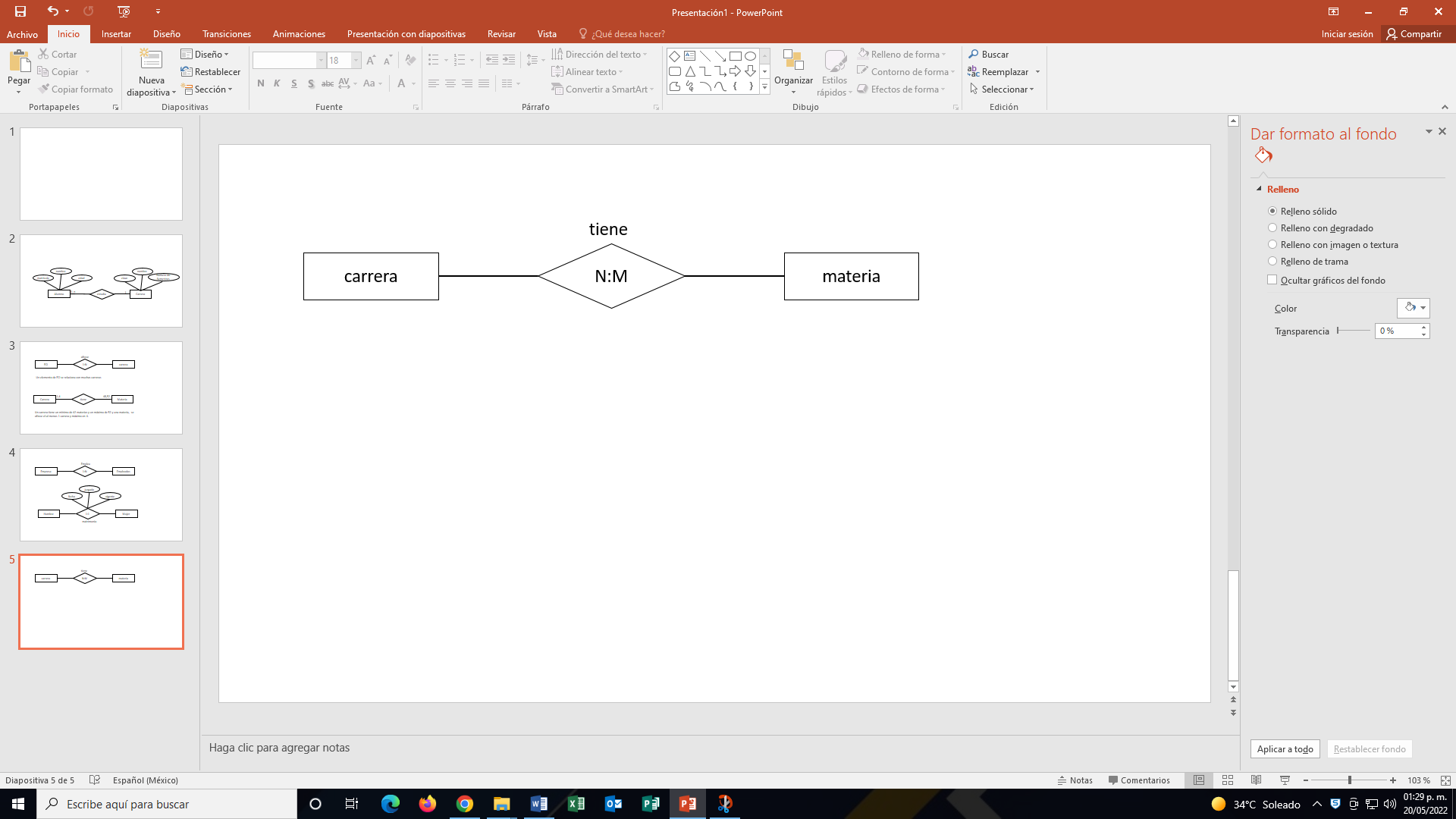
Para este ejemplo: la tabla empleado queda así:

Tabla empleado

|  |  |  |  |
| --- | --- | --- | --- |
| rfc | nombre | especialidad | empresa |
| Bp123465 | Berta Perez | soldador | Alu1234 |
| Au785423 | Ana Uc | cocinera | Pc7982 |

Esta tabla dice que Berta trabaja en aluminios del Carmen y que Ana trabaja en panuchos chito.

Ahora suponga una relación muchos a muchos como esta:



Esta relación dice que en una carrera se imparten muchas materias y que una materia se imparte en muchas carreras (por ejemplo, la materia de tics se da en todas las carreras de la UNACAR).

Para convertir a tablas:

Primero convertimos a tablas las entidades:

Tabla carrera

|  |  |  |  |
| --- | --- | --- | --- |
| clave | nombre | Creditos | certificada |
| 9867 | ISC | 110 | si |
| 2345 | ICO | 115 | si |

Tabla materia

|  |  |  |  |
| --- | --- | --- | --- |
| clave | nombre | horasSemena | creditos |
| M01 | tic | 5 | 6 |
| M02 | Algebra I | 6 | 7 |

Ahora la relación la convertimos a tabla

Tabla carrera\_materia

|  |  |
| --- | --- |
| Clave\_carrera | Clave\_materia |
| 9867 | M01 |
| 9867 | M02 |

## Objetivo de la práctica 5

Obtener el conjunto de tablas de una base de datos a partir de un modelo ER.

## Materiales de la práctica 5

* Computadora
* Internet
* Suite de programas de oficina

## Desarrollo de la práctica 5

1. Tome el modelo ER diseñado para el proceso de gestión académica en la FCI.
2. Convierta el diseño en tablas.
3. Marque cuales son las llaves primarias de cada tabla.

## Entregables de la práctica 5

1. Archivo pdf con el diagrama de tablas
2. Archivo fuente con el diagrama de tablas

# 6.- Modelo relacional.

Una relación es una regla que asocia elementos de un conjunto en otro, es un subconjunto del producto cartesiano de dos o más conjuntos. Por ejemplo sea el conjunto de géneros femenino y masculino A={f,m}, y el conjunto de personas B={Ana, Bety, Carlos, Enrique}, el producto cartesiano AxB es:

{ (f,Ana), (f,Bety), (f,Carlos), (f,Enrique),

(m,Ana), (m,Bety), (m,Carlos), (m,Enrique) }

Estas son todas las combinaciones de los elementos de A y de B, Sin embargo, supongamos que queremos identificar las mujeres, debemos aplicar la ***regla*** que dice R=” Elemento de B que es de sexo femenino”, el conjunto resultante es:

{(f,Ana), (f,Bety)}

Y son las 2 únicas parejas que cumplen la regla

Tenemos una regla de asociación, observe que el conjunto generado por la regla es efectivamente un subconjunto del producto cartesiano**. La regla es una relación** y decimos que del conjunto A y B representamos la relación R como: {(f,Ana), (f,Bety)}.

Las relaciones se muestran en forma de tabla:

Tabla 6.1 Relación R

|  |  |
| --- | --- |
| Sexo | Nombre |
| F | Ana |
| f | Bety |

Decimos entonces que **una tabla es una relación**, y **un conjunto de relaciones conforman la base de datos**.

## 6.1.- Relaciones y dominios.

Cada columna o campo de la relación debe tener un dominio y el dominio es el fondo de valores de donde obtenemos los datos en la tabla 6.1 el dominio del campo sexo es el conjunto A={f,m} y el dominio del campo Nombre es b={ Ana, Bety, Carlos, Enrique }, el dominio depende del problema que estemos modelando por ejemplo si estamos diseñando una base de datos (DB) para llevar las calificaciones de los alumnos de la licenciatura en tecnologías de información (LTI) de la FCI- UNACAR, el dominio del campo nombre será todos los nombres de alumnos inscritos a LTI.

Cada renglón de la relación es llamado **tupla** o registro.

## 6.2.- reglas de integridad relacional.

En el modelo relacional para cada relación (tabla) en la base de datos cada registro debe ser único, es decir los datos no deben repetirse, hay que aclarar que hablamos del registro completo, es decir si tengo una tabla *empleados* con *rfc* y *nombre*.

Tabla 2.2 Tabla Empleados

|  |  |
| --- | --- |
| rfc | Nombre |
| DITR750310 | Rosa Díaz Tun |
| DITR831212 | Rosa Díaz Tun |

Tenemos dos registros (tuplas) diferentes pues, aunque el nombre se repite todo el renglón es diferente pues con que un solo campo sea diferente se toma como un registro diferente. Con esto llegamos al concepto de llave; un campo o conjunto de campo es llave siempre y cuando determine de manera única a la tupla.

**Determinante**. Si el valor de campo X de la relación está asociado a un solo valor Y en la relación (o sea para un valor x del campo X, se encuentra un único valor y del campo Y), decimos que X determina a Y, o qué Y es funcionalmente dependiente de X.

X🡪Y

Por ejemplo sea ***X*** el campo rfc y ***Y*** el campo Nombre en la relación empleados, para un valor X (o sea para un rfc), se puede encontrar uno y solo un nombre. Ejemplo si yo tengo el rfc=” DITR750310” ¿cuantos nombres encuentro con ese rfc? respuesta= uno solo, el nombre del registro 1 en la tabla anterior tiene ese rfc. En cambio, el campo nombre no determina a rfc, pues si yo tengo el nombre “Rosa Díaz Tun”, cuantos rfc encuentro que tengan ese nombre respuesta= dos rfc tienen ese nombre por lo que el campo nombre no es un determinante del campo rfc.

Un campo es llave si determina a todos los demás campos en la tabla, es decir si todos los demás campos son funcionalmente dependientes de el.

### 6.2.1.- Calves primarias.

Una llave primaria es un campo o conjunto de campos que ***determina*** de manera única a todos los demás atributos (campos) en la tupla (renglón) de la relación. Pueden existir varias claves candidatas a ser clave primaria, en este caso se elige una como clave primaria y las demás son clave candidata. Por ejemplo si la tabla Alumno tiene los atributos (rfc, matricula, nombre, carrera) tenemos 2 claves candidatas pues (rfc y matricula) son irrepetibles para cada alumno y por lo tanto pueden ser ambas una llave, el diseñador debe elegir una de ellas para ser llave primaria y la otra se convierte en llave candidata.

### 6.2.2.- Regla de integridad de las entidades.

Para garantizar el acceso consistente a la información existen reglas d integridad:

1. Debe existir una llave en todas las tablas de la relación.
2. Los valores del campo (o campos llaves) no pueden ser nulos.
3. Si un campo en la relación hace referencia a un campo llave en otra relación, el campo en la otra relación debe existir, esto se conoce como **regla de integridad referencial**.

### 6.2.3.- Claves foráneas.

Una clave o llave foránea es un campo que hace referencia a otro campo que es llave en otra relación. Por ejemplo:

 Tabla Alumnos

|  |  |  |
| --- | --- | --- |
| Matricula | Nombre | Carrera |
| 120989 | Angel Coba | 01 |
| 110067 | Rosa Celeste | 02 |
| 107856 | Pedro Saenz | 03 |

Tabla Carreras

|  |  |  |
| --- | --- | --- |
| Clave | Carrera | Semestres |
| 01 | Licenciatura en Informática | 8 |
| 02 | Ingeniería en sistemas Computacionales | 8 |
| 03 | Ingeniería en computación | 9 |

En las tablas *Alumnos* y *Carreras* se muestran sus campos llave (*matricula* y *clave* respectivamente) con una llave encima del campo, si vemos el campo carrera de la tabla *Alumnos* tiene datos que son valores de *llave* en la tabla *Carreras*, a esto se denomina llave ajena, el campo *Carrera* es una llave ajena pues contiene valores llave, pero no en su propia tabla. La regla de integridad referencial dice que para cada valor del campo *Carrera*, debe existir el correspondiente en el campo llave de la tabla *Carreras*.

## 6.3.- Álgebra relacional.

### 6.3.1.- Operaciones con conjuntos.

**Unión de conjuntos.**

Es la operación que nos permite unir dos o más conjuntos para formar otro conjunto que contendrá a todos los elementos que queremos unir, pero sin que se repitan. Es decir, dado un conjunto A y un conjunto B, la unión de los conjuntos A y B será otro conjunto formado por todos los elementos de A, con todos los elementos de B sin repetir ningún elemento. El símbolo que se usa para indicar la operación de unión es el siguiente: ∪. Cuando usamos diagramas de Venn, para representar la unió de conjuntos, se sombrean los conjuntos que se unen o se forma uno nuevo. Luego se escribe por fuera la operación de unión.

Ejemplo 1.

Dados dos conjuntos A={1,2,3,4,5,6,7,} y B={8,9,10,11}

la unión de estos conjuntos será A∪B={1,2,3,4,5,6,7,8,9,10,11}.

**Intersección de conjuntos.**

Es la operación que nos permite formar un conjunto, sólo con los elementos comunes involucrados en la operación. Es decir, dados dos conjuntos A y B, la de intersección de los conjuntos A y B, estará formado por los elementos de A y los elementos de B que sean comunes, los elementos no comunes A y B, será excluidos. El símbolo que se usa para indicar la operación de intersección es el siguiente: ∩

Ejemplo 1.

Dados dos conjuntos A={1,2,3,4,5} y B={4,5,6,7,8,9} la intersección de estos conjuntos será A∩B={4,5}.

**Complemento de un conjunto.**

Es la operación que nos permite formar un conjunto con todos los elementos del conjunto de referencia o universal, que no están en el conjunto. Es decir, dado un conjunto A que está incluido en el conjunto universal U, entonces el conjunto complemento de A es el conjunto formado por todos los elementos del conjunto universal pero sin considerar a los elementos que pertenezcan al conjunto A. En esta operación el complemento de un conjunto se denota con un apostrofe sobre el conjunto que se opera, algo como esto A' en donde el el conjunto A es el conjunto del cual se hace la operación de complemento.

Ejemplo 1.

Dado el conjunto Universal U={1,2,3,4,5,6,7,8,9} y el conjunto A={1,2,9}, el conjunto A' estará formado por los siguientes elementos A'={3,4,5,6,7,8}.

### 6.3.2.- Proyección, SELECCIÓN, producto cartesiano, diferencia.

**Proyección de conjuntos.**

Sea R un conjunto de atributos R={a1, …, an}, una proyección p(1,3,7) en R, una proyección es el resultado de filtrar los atributos de la relación R a los que indica la proyección en este caso:

P={a1,a3,a7}

**Producto cartesiano**

El producto cartesiano de dos conjuntos A, B es el resultado de hacer todas las posibles combinaciones de los elementos de A con los elementos de B.

Ejemplo: sea a={2,3} y B={a,b}

El producto cartesiano de AxB es={(2,a),(2,b),(3,a),(3,b)}

**Diferencia de conjuntos.**

Es la operación que nos permite formar un conjunto, en donde de dos conjuntos el conjunto resultante es el que tendrá todos los elementos que pertenecen al primero, pero no al segundo. Es decir, dados dos conjuntos A y B, la diferencia de los conjuntos entra A y B, estará formado por todos los elementos de A que no pertenezcan a B. El símbolo que se usa para esta operación es el mismo que se usa para la resta o sustracción, que es el siguiente:

Ejemplo 1.

Dados dos conjuntos A={1,2,3,4,5} y B={4,5,6,7,8,9} la diferencia de estos conjuntos será A-B={1,2,3}.

## 6.4.- Cálculo relacional.

### 1.- La operacionde seleccion.

Suponga la tabla alumno

|  |  |  |
| --- | --- | --- |
| matricula | nombre | carrera |
| 212345 | Juana Rioz | isc |
| 217865 | Ana Uc | isc |
| 223454 | Pedro Chan | ico |
| 123465 | Tere paz | lti |

Suponga que queremos seleccionar todas las tuplas de isc, la operación es:

**Select \* from alumno where carrera=”isc”**

El resultado es:

|  |  |  |
| --- | --- | --- |
| matricula | nombre | carrera |
| 212345 | Juana Rioz | isc |
| 217865 | Ana Uc | isc |

### 2 La operación de proyección

Suponga que queremos filtrar algunos campos (atributos de la relación) y que queremos solo la matrícula y el nombre:

**Select matricula, nombre from alumno**

|  |  |
| --- | --- |
| matricula | nombre |
| 212345 | Juana Rioz |
| 217865 | Ana Uc |
| 223454 | Pedro Chan |
| 123465 | Tere paz |

En una proyección puede cambia el nombre de los campos así:

**Select matricula as x, nombre as y from alumno**

|  |  |
| --- | --- |
| x | y |
| 212345 | Juana Rioz |
| 217865 | Ana Uc |
| 223454 | Pedro Chan |
| 123465 | Tere paz |

Es la misma operación, pero estamos cambiando el nombre de los campos.

### 3 La operación de interseccion

Suponga que tiene las tablas

Alumno

|  |  |  |
| --- | --- | --- |
| matricula | nombre | carrera |
| 212345 | Juana Rioz | isc |
| 217865 | Ana Uc | isc |

Carrera

|  |  |  |
| --- | --- | --- |
| clave | nombre | gestor |
| isc | Ingeniería es sistemas | Juan Carlos canto |
| ico | Ingeniería en computación | Fernando Sanchez |

Y quiero una lista de los alumnos con los nombres de su gestor:

**Select alumno.nombre, carrera.gestor from alumno inner join carrera on alumno.carrera=carrera.clave**

Resultado:

|  |  |
| --- | --- |
| nombre | gestor |
| Juana Rioz | Juan Carlos Canto |
| Ana Uc | Juan Carlos Canto |

## Objetivo de la práctica 6.1

Aplicar operaciones del algebra relacional.

## Materiales de la práctica 6.1

* Computadora
* Internet
* Suite de programas de oficina

## Desarrollo de la práctica 6.1

1. Realice los ejercicios que se indican a continuación:

|  |
| --- |
| Ejercicios. |
| 1. Para las siguientes tablas, proponga una llave primaria:    1. Tabla consulta: campos (clavePaciente, claveMedico, sitomas, diagnóstico, tratamiento, fecha).    2. Tabla envio, campos (claveCliente, claveProducto, cantidad, costo, fecha) |
| 1. Para las tablas: venta con campos (numeroVenta, fecha, total), y la tabla detalleVenta con campos (numeroVenta, claveProducto, catidad, subtotal), realice lo siguiente:    1. Obtenga una lista para los productos con clave “23” que se vendieron en la venta número “2”.    2. Obtenga una lista de todos los productos y cantidades para la venta 4.    3. Indique cual sería la llave primaria de ambas tablas. |
| 1. En la liga: <https://www.w3schools.com/sql/trysql.asp?filename=trysql_asc> realice lo siguiente:    1. Del lado derecho puede ver las tablas disponibles: (**customers**, **categoryes**, **employees**, …) elija la tabla **customers**. A continuación, en el área llamada **SQL Statement**: podrá ejecutar consultas **sql** sobre las tablas mencionadas. Como ejemplo ejecute el comando: (**select \* from orders**) verá el resultado de la sentencia en el área **Result**: a continuación:    2. Haga una proyección de los campos **customerid** y **customername** de la tabla **customers**.    3. Muestre las tablas productos y después la tabla **categoryes**    4. Muestre los datos de los productos incluyendo nombre de la categoría(**categoryname**) y descripción (**description**)    5. Repita el inciso e, pero solo para los de precio (**price**) menor o igual a 10    6. Repita el inciso e pero proyecte los campos (**productname, categoryname y description**)    7. Guarde las sentencias que uso en cada inciso para los entregables de esta práctica. |
| 1. Repita el inciso 3-f pero los nombres de los campos que muestre la consulta deben ser: (**producto, categoría y descripción**) |

## Entregables de la práctica 6

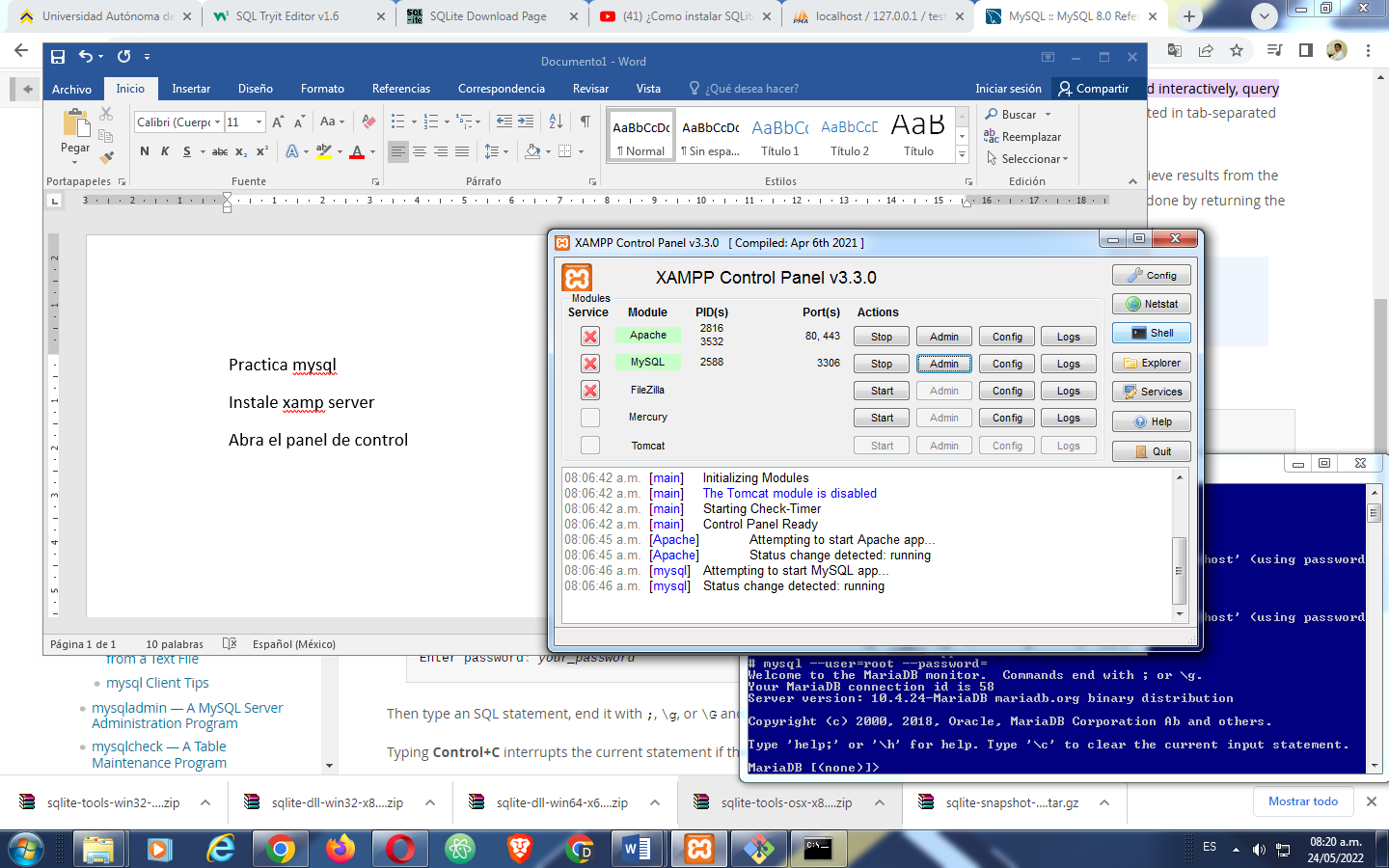
Archivo pdf con las sentencias sql de los ejercicios.

## 6.5.- Uso de un DBMS en linea de comandos.

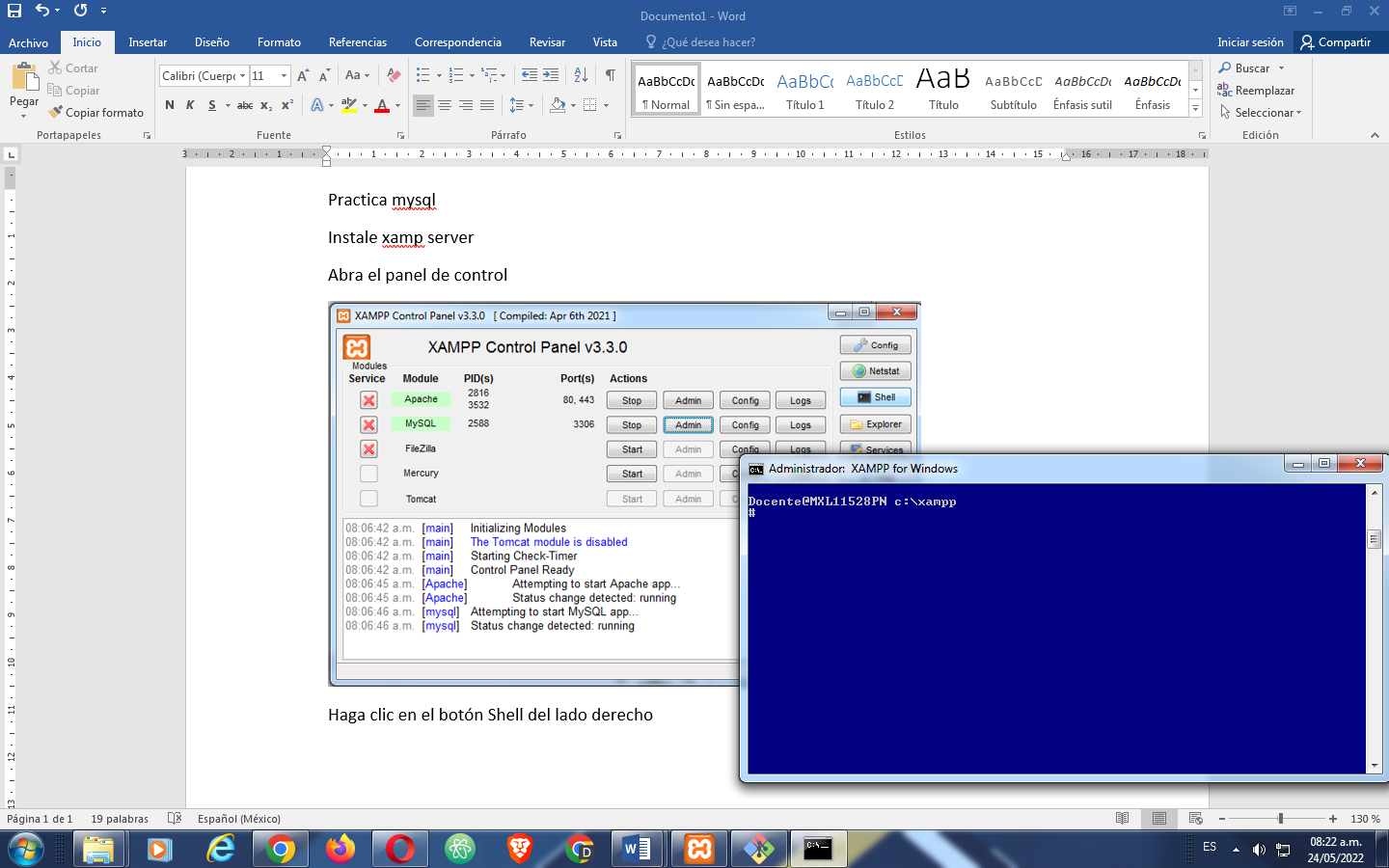
### 1.- Instalar XAMP

Para esta práctica deberá:

1. Instalar Xamp server.
2. Abra el panel de control de Xamp.



1. Haga clic en el botón Shell del lado derecho, verá:



### Conéctese al DBMS (database manager system) con el comando:

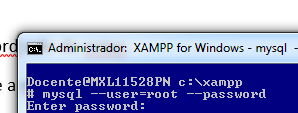
#mysql --user=root –-password

Mostrará el mensaje:

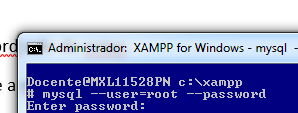
Enter password:

Si instalo el ***Xamp*** sin passwoprd solo de enter.

Por ejemplo, para conectarme a ***mysql*** escribí en la consola de comando:

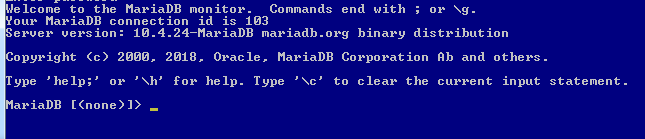


En la línea que pide “Enter password”:



Solo presione la tecla <enter>.

Una vez conectado a su DBMS verá:



Que indica que ya se conectó a su DBMS. El mensaje ***[(none)]*** indica que no está usando ninguna base de datos de su DBMS.

### Comandos

Con el comando ***help*** muestra los diferentes comandos de ***mysql Shell***.

***>help;***

El siguiente comando nuestra la fecha del sistema:

***>select curdate();***

Si quiere mostrar las bases de datos existentes pruebe:

***>Show datbases;***

El comando ***quit*** sale de mysql Shell

1. Crear una base de datos

Crear una base de datos llamada prueba:

**>create dabase if not exists prueba;**

Poner en uso una base de datos:

**>use prueba;**

****

Observe que después del comando use se muestra [prueba]>, que indica la base de datos con que está trabajando actualmente.

Mostrar las tablas de la base de datos:

**>show tables;**

Crear tabla de base de datos; crearemos la tabla carrera con campos (id, nombre, créditos) y llave primaria id:

**>create table if not exists carrera (id int, nombre varchar(50), creditos int, primary key(id));**

Como la base de datos en uso es prueba, esta tabla se creará dentro de esa base de datos.

Insertar tuplas en la tabla:

**>insert into carrera (id, nombre, creditos) values(1,”isc”,110);**

Mostrar tuplas de la tabla:

**>select \* from carrera;**

Suponga que quiere hacer una proyección de la tabla y mostrar solo el campo nombre:

**>select nombre from carrera;**

Suponga que quiere una tabla alumno con campos (***id, nombre y idcarrera***); donde id es la llave primaria y es auto incrementable y ***idcarrera*** es una llave ajena referida a la tabla carrera, es decir el contenido de este campo debe corresponde con una tupla de la tabla carrera:

**>create table if not exists alumno (**

**id int not null auto\_increment,**

**nombre varchar(50),**

**idcarrera int,**

**primary key(id),**

**foreign key(idcarrera) references carrera(id)**

**);**

La línea: ***foreign key(idcarrera) references carrera(id)***, indica que el campo ***idcarrera*** es una llave ajena que hace referencia al campo ***id*** de la tabla ***carrera***. Como en el modelo relacional, la base de datos debe respetar la integridad referencial, al insertar un dato en el campo ***idcarrera*** este debe corresponder con un dato existente en el campo ***id*** de la tabla ***carrera***, en caso contrario marcará un error.

**Insertar datos con un campo auto incrementable.**

Recordemos que ya creamos una tabla ***carrera*** y le insertamos un registro con ***id***=1 y que la tabla ***alumno*** tiene una llave foránea a la tabla carrera llamada (***idcarrera***), por lo que, al insertar un dato, el valor del campo (***idcarrera***) de la tabla ***alumno***, debe corresponder con un valor en el campo ***id*** de la tabla carrera, entonces hacemos:

**>insert into alumno (nombre, idcarrera) values(“juan uc”, 1);**

Aquí debemos respetar lo siguiente:

1) no se insertará dato en el campo id de la tabla alumno pues este es ***auto\_increment***, por lo que no se indica en la lista de campos a insertar ni en la lista de datos a insertar en la sentencia ***insert***.

2) como hay una llave foránea (o llave ajena), al insertar un dato en la tabla ***alumno***, el dato a insertar en el campo ***idcarrera*** que corresponde a la llave foránea, debe existir en la llave primaria de la tabla asociada en el ***foreign key***.

## Objetivo de la práctica 6.2.

Implementar y consultar una base de datos usando sentencias SQL.

## Materiales de la práctica 6.2

* Manual de prácticas
* Internet
* Navegador
* Suite de oficina

## Desarrollo de la práctica 6.2

1 Deberá crear la base de datos provpar.

2 Deberá: crear una tabla llamada **tipo\_parte** con los campos (id int, nombre varchar(50)), el id será llave primaria auto incrementable, inserte los datos:

(1, “material eléctrico”)

(2, “material plomería”)

(3, “material carpinteria”)

2 Cree una tabla **partes** con campos (id int, nombre varchar(50), stock\_minimo int, stock int, stock\_maximo int, costo\_compra, costo\_venta, tipo\_parte llave foránea a la tabla tipo\_parte),

Inserte los valores:

(1, “tornillo ½”, 20, 20, 100, 1,2, 3)

(1, “tornillo 3/8”, 20, 20, 100, 1,2, 3)

(1, “tuerca ½”, 20, 20, 100, 3,4, 3)

(1, “tuerca 3/8”, 20, 20, 100, 3,4, 3)

(1, “martillo 10 oz”, 10, 100, 100, 20,30,2)

(1, “pinzas aisladas chicas”, 10, 100, 100, 20,30,1)

(1, “pinzas de corte chicas”, 10, 100, 100, 20,30,1)

3 Obtenga las siguientes consultas con sentencias sql en la línea de comando:

1. Lista de partes de carpintería
2. Lista de partes de electricidad
3. Lista de nombres de parte junto con los nombres de su tipo de parte
4. Lista de partes que están en el stock mínimo
5. Lista del margen de ganancia de cada parte

# 7.- Diseño de bases de datos modelo relacional.

## 1.- El enfoque intuitivo.

En este punto deberá analizar la narrativa de su problema e identificar las entidades (sustantivos) que en él se mencionan, Haga una lista con las entidades, revise la lista y depure, elimine las que no tengan que ver con el problema, las entidades resultantes se convertirán en tablas. A las tablas resultantes asócieles atributos según su problema.

Ejemplo suponga el siguiente problema:

Tabla 2.4.1.1

|  |
| --- |
| Narrativa de la empresa Panuchos S.A. |
| La empresa **Panuchos del Playón S.A**., dedicada a la elaboración de panuchos en todo la república, tiene varias gerencias como son, Compras, Ventas, Tratamiento de información, Almacén y Transportes, debido a problemas sindicales esta empresa opto desde hace 15 años por la contratación del 90% de su personal a través de OutSourcing, para lo cual el consejo de administración periódicamente destina un conjunto de fondos de capital asociados a un rubro específico cada uno, por ejemplo existe un fondo para desarrollo de software del cual se extraen las partidas de capital necesarias para el pago de salarios y equipos de cómputo para el rubro de desarrollo de software para la empresa. Los fondos de capital pueden ser utilizados si existen partidas autorizadas, cada partida se asocia directamente a un contrato de algún rubro outsourcing y tiene un monto tope. Los contratos junto con el monto asociado se aprueban después de una evaluación, en el consejo de administración, y pueden ser usados hasta que el director general los autorice. La empresa desea el desarrollo de un sistema que controle a) las partidas con su rubro y su cantidad, su monto tope, b) los contratos con su fecha de inicio, fecha de final, y partida, c) los datos de las empresas de outsourcing como son, clave, razón social, dirección, teléfono, representante legal y el contrato y partida al que están sujetos, cada empresa puede tener varios contratos, por cada contrato se asigna un supervisor de la empresa que revise y autorice la utilización de los montos de capital en las partidas, d) datos de los representantes legales de las empresas como nombre, teléfono, e-mail. El supervisor recibe las solicitudes de capital de las compañías de outsourcing, las evalúa y autoriza en su caso, los datos que la compañía debe enviar al solicitar montos de capital son: clave de la compañía, contrato, partida, monto solicitado, destino de los fondos, desglose de la utilización de los recursos. Una vez recibida la solicitud de fondos el supervisor genera un documento de autorización de capital que contiene la clave de la solicitud de fondos que hizo la empresa, el monto autorizado y el desglose del destino autorizado de los fondos. |

En el ejemplo de la tabla 2.4.1.1 se muestra marcados en verde las entidades y en azul los atributos. Hacemos una lista con las posibles entidades:

|  |
| --- |
| Entidades |
| empresa **Panuchos del Playón S.A** |
| panuchos |
| Gerencia compras |
| Gerencia ventas |
| Gerencia tratamiento de Información |
| Gerencia Almacén |
| Gerencia Transportes |
| Consejo de administración |
| Fondos de capital |
| Partidas de capital |
| Rubro |
| Salarios |
| Equipos de computo |
| Contrato de outsourcing |
| Moto tope |
| Consejo de administración |
| Director general |
| Empresa de outsourcing |
| Supervisor de la empresa |
| Solicitudes de capital |
| autorización de capital |

Analizando el problema vemos que las entidades que nos piden controlar solo son las que marcamos en amarillo y las demás las eliminamos pues no son pertinentes para nuestro problema particular.

Para las tablas resultantes revise sus asociaciones que pueden ser 1 a 1, uno a muchos, o muchos a muchos, por ejemplo:

* Un alumno se asocia con una carrera (1-m).
* Un hombre se casa con una mujer (1-1).
* Un alumno toma varias materias y una materia es tomada por muchos alumnos (m-m).

Resuelva las asociaciones como sigue:

1. (1-1) y (m-m) creará una nueva tabla que incluya las llaves primarias de ambas tablas (La asociación (1-1) es un caso particular de una relación (m-m) por lo tanto se resuelve igual).
2. (1-m) creará un nuevo campo en la tabla muchos que será una llave ajena y contendrá la llave primaria de la tabla 1.

Ejemplos:

Ejemplo de asociación 1-m (uno a muchos), suponga las tablas:

|  |  |
| --- | --- |
| Tabla Alumno | Tabla Carrera |
| |  |  |  | | --- | --- | --- | | Matricula | Nombre | Sexo | | 110967 | Juan Puc | M | | 118967 | Rosa Chi | F | | 129087 | Javier Mass | M | | |  |  |  | | --- | --- | --- | | Clave | Nombre | Gestor | | Lti2011 | Licenciatura en TI | 779 | | Ico2010 | Ingeniería en Computación | 876 | | Isc2010 | Ingeniería en sistemas C. | 398 | |

Suponga además que cada alumno puede estar inscrito en una carrera y en cada carrera puede haber muchos alumnos inscritos, por lo que es una relación 1-m de Alumno – Carrera. Esto se resuelve agregando un campo más en la tabla muchos que tenga la llave primaria de la tabla 1 y este nuevo campo será una llave ajena:

|  |  |
| --- | --- |
| Tabla Alumno | Tabla Carrera |
| |  |  |  |  | | --- | --- | --- | --- | | Matricula | Nombre | Sexo | Carrera | | 110967 | Juan Puc | M | Ico2010 | | 118967 | Rosa Chi | F | Ico2010 | | 129087 | Javier Mass | M | Isc2010 | | |  |  |  | | --- | --- | --- | | Clave | Nombre | Gestor | | Lti2011 | Licenciatura en TI | 779 | | Ico2010 | Ingeniería en Computación | 876 | | Isc2010 | Ingeniería en sistemas C. | 398 | |

Para garantizar un buen diseño, después de resolver las asociaciones, a las tablas resultantes aplíqueles normalización, las formas normales 1,2 y 3.

## Objetivo de la práctica 7.1.

Implementar y consultar una base de datos usando sentencias SQL.

## Materiales de la práctica 7.1.

* Manual de prácticas
* Internet
* Navegador
* Suite de oficina

## Desarrollo de la práctica 7.1.

En su equipo de trabajo, elija una problemática de gestión de datos de su ámbito, aplique el modelo intuitivo para crear la base de datos que soporte la información.

## Entregable de la práctica 7.1.

Archivo pdf con su propuesta de base de datos junto con la narrativa de donde genero la propuesta.

## 2.- Normalización.

Toda base de datos relacional debe cumplir con las formas normales.

### 1.- Primera forma normal.

Primera forma normal. Todos los atributos o campos deben ser atómicos.

Ejemplo, la tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla Alumno que no está en 1NF | | | |
| RFC | Nombre | Materia | Sexo |
| PEL021002 | Juan Perez López | Cálculo, Algebra, Lógica | M |
| PEL021501 | Angel Ríos Luna | Redes, Cálculo | M |
|  |  |  |  |

Esta tabla no tiene valores atómicos en el campo materia, que sea atómico quiere decir que en cada campo existe solo un valor de su dominio de valores, por ejemplo: para el campo rfc el dominio que es el conjunto o fondo de valores permitido para ese campo es: todos los rfc existentes, por ejemplo, la secuencia de caracteres: “fohj640421” es un valor del dominio sin embargo la secuencia “5634dfre89” no pertenece al dominio. Cada campo tiene su dominio, para el nombre el dominio es el conjunto de nombres de personas, para materia el conjunto de nombres de materias.

Como vemos en la tabla anterior en el campo materia no hay valores atómicos 8es decir que se tome un solo valor del dominio) ya que en el ejemplo en el primer registro estamos diciendo que juan Pérez López cursa las materias de cálculo, algebra y lógica. Pero en el campo materia no estamos almacenando un valor si no varios.

Si esta tabla la representamos como sigue ahora si serán valores atómicos y ahora es una tabla que está en primera forma normal (1NF).

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla Alumno en 1FN | | | |
| RFC | Nombre | Materia | Sexo |
| PEL021002 | Juan Perez López | Cálculo | M |
| PEL021002 | Juan Perez López | Algebra | M |
| PEL021002 | Juan Perez López | Lógica | M |
| PEL021501 | Angel Ríos Luna | Redes | M |
| PEL021501 | Angel Ríos Luna | Calculo | M |

### 2.- Segunda forma normal.

Una tabla esta es segunda forma normal 2NF, si está en primera forma normal y todos los atributos no clave (no llave), dependen totalmente de la llave primaria. Ejemplo:

Suponga la tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla: Calificaciones | | | |
| Matricula | ClaveMateria | Nombre | Calificacion |
| 110967 | MT01 | Juan Puc | 75 |
| 110967 | PR01 | Juan Puc | 80 |
| 118967 | MT01 | Rosa Chi | 90 |
| 118967 | PR01 | Rosa Chi | 50 |
| 118967 | ED01 | Rosa Chi | 100 |

Esta tabla no es 2NF, para saber por qué veamos sus dependencias funcionales: primeramente, note que la llave primaria de la tabla no es un solo campo si no 2 (matricula y claveMateria), razón: la matricula se repite y una llave debe ser única, la claveMateria también se repite, pero tomando ambas el dúo no se repite por que para un alumno (determinado por una matrícula) y una materia (determinada por una claveMateria) existe solo una calificación. Por lo que tenemos las siguientes dependencias funcionales:

1. Matricula,Matria -> Nombre
2. Matricula,Matria -> Calificacion
3. Matricula -> nombre

Vemos que la tercera dependencia indica que el nombre depende funcionalmente solo de la matricula o sea que la matricula por si sola determina el nombre del alumno, por lo que nombre siendo un atributo no llave no depende de toda la llave primaria y por esto no está en 2NF, solución:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Tabla: Calificaciones | | | | Matricula | ClaveMateria | Calificacion | | 110967 | MT01 | 75 | | 110967 | PR01 | 80 | | 118967 | MT01 | 90 | | 118967 | PR01 | 50 | | 118967 | ED01 | 100 | | |  |  | | --- | --- | | Tabla: Alumnos | | | Matricula | Nombre | | 110967 | Juan Puc | | 118967 | Rosa Chi | |

### 3.- forma normal de Boyce codd.

Una tabla está en forma normal de Boyce Codd BCNF si está en 2NF y además todo determinante es una clave candidata.

Supongamos que necesitamos saber lo que un alumno foráneo gasta en transportarse desde su ciudad de origen, podemos tener la tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabla Alumno en 2FN | | | | |
| Matricula | Nombre | Sexo | Ciudad | Costo |
| 128967 | Juan Perez López | M | Frontera | 50 |
| 112345 | Rosa Chan | F | Sabancuy | 60 |
| 119867 | Rafael Chi | M | Champoton | 100 |
| 101132 | Angel Cobos | M | Frontera | 50 |
| 125634 | Erika Uc | F | Campeche | 145 |

Esta tabla no es BCNF, veamos la razón, obtengamos los determinantes, como vemos matricula determina a nombre, sexo ciudad y costo.

Matricula -> Nombre, Sexo, Ciudad, Costo

Vemos también que ciudad determina el costo, pues el costo del transporte del alumno depende de la ciudad de origen.

Ciudad -> Costo

Pero Ciudad solo determina al costo y no determina a los demás atributos por lo tanto tenemos un determinante que no es llave candidata y la tabla no es BCNF. La solución es dividir en dos tablas así:

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla Alumno en BCNF | | | |
| Matricula | Nombre | Sexo | Ciudad |
| 128967 | Juan Perez López | M | Frontera |
| 112345 | Rosa Chan | F | Sabancuy |
| 119867 | Rafael Chi | M | Champoton |
| 101132 | Angel Cobos | M | Frontera |
| 125634 | Erika Uc | F | Campeche |

|  |  |
| --- | --- |
| Tabla Costo en BCNF | |
| Ciudad | Costo |
| Frontera | 50 |
| Sabancuy | 60 |
| Champoton | 100 |
| Campeche | 145 |

## Objetivo de la práctica. 7.2.

Aplicar los conceptos de llave, determinante, formas normales al diseño de sus bases de datos.

## materiales de la práctica 7.2.

* Computadora
* Internet
* Suite de programas de oficina

## desarrollo de la práctica. 7.2.

1. Para la narrativa que en equipo definió en el modelo intuitivo obtenga los siguiente:
   1. Determinantes de todas las tablas y diagramas de dependencia funcional.
   2. Llaves primarias de todas las tablas.
   3. Llaves ajenas donde aplique.
   4. Aplique las formas normales 1,2 y BCNF.
   5. Diagrama final de diseño de su BD.
2. Para la narrativa de panuchos, repita el ejercicio anterior.
3. Para la narrativa de provPar repita el ejercicio anterior.

## entregables de la práctica. 7.2.

* Archivo pdf con el resultado de su actividad donde se muestren:
  + Las narrativas.
  + Los diagramas de dependencia funcional.
  + Las llaves primarias y ajenas.
  + El diseño final de sus bases de datos

# 8.- Sistemas manejadores de bases de datos (dbms).

## 1.- Data base manager system.

Un sistema manejador de bases de datos (DBMS) es un sistema de software que permiten crear y administrar la base de datos, permiten la creación y modificación de la base, administración del espacio, consultas, administración de usuarios y permisos, procedimientos almacenados y manejo de restricciones a la base. Existen diferentes manejadores de base de datos con más o menos prestaciones, están Oracle, Informix, SQL Server, MySql, y tenemos también un conjunto de manejadores ligeros con pocas prestaciones, pero con pocos requerimientos de software como Acces y SqLite, también podemos dividir a los manejadores en libres y con pago de licencia, entre los libres tenemos MySql, JavaDB (antes Derby) y SQLite.

Para estas notas usaremos MySql, vea en la presentación “instalacionWampyHeidi.pptx” como apoyo para la instalación de una versión de MySQL (wamp) y un administrador (Heidi Sql).

## 2.- Creación de la base de datos.

Vea el video “<https://www.youtube.com/watch?v=40YYmufKXus>” que muestra un ejemplo de creación de base de datos.

## 3.- Catalogo del sistema.

El catálogo del sistema es un conjunto de tablas que muestran la información que el DBMS usa para administrar las bases de datos.

## 4.- Vistas.

Una vista es una tabla virtual, se construye a partir de una consulta y los usuarios de la base de datos la ven como una tabla más pero internamente cuando es consultada, se construye como una sentencia select.

## 5.- Sql.

SQL es el lenguaje de consulta estructurado que nos permite obtener información de la base de datos, la sentencia principal es SELECT.

## 6.- Las doce reglas DE. CODD.

Edgar Cood es considera el padre de las bases de datos relacional. En 1969 Edgar Codd inventó el modelo relacional, el modelo de bases de datos más usado hoy en día.

Codd se percató de que existían bases de datos en el mercado las cuales decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar estas tablas literalmente normalizadas; entonces Cood publicó 12 reglas que un verdadero sistema relacional debería tener, aunque en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse «más relacional» cuanto más siga estas reglas.

– Regla 0: el sistema debe ser relacional, base de datos y administrador de sistema. Ese sistema debe utilizar sus facilidades relacionales (exclusivamente) para manejar la base de datos.

– Regla 1: la regla de la información, toda la información en la base de datos es representada unidireccionalmente, por valores en posiciones de las columnas dentro de filas de tablas. Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera: con valores en tablas.

– Regla 2: la regla del acceso garantizado, todos los datos deben ser accesibles sin ambigüedad. Esta regla es esencialmente una nueva exposición del requisito fundamental para las llaves primarias. Dice que cada valor escalar individual en la base de datos debe ser lógicamente direccionable especificando el nombre de la tabla, la columna que lo contiene y la llave primaria.

– Regla 3: tratamiento sistemático de valores nulos, el sistema de gestión de base de datos debe permitir que haya campos nulos. Debe tener una representación de la «información que falta y de la información inaplicable» que es sistemática, distinto de todos los valores regulares.

– Regla 4: catálogo dinámico en línea basado en el modelo relacional, el sistema debe soportar un catálogo en línea, el catálogo relacional debe ser accesible a los usuarios autorizados. Es decir, los usuarios deben poder tener acceso a la estructura de la base de datos (catálogo).

– Regla 5: la regla comprensiva del sublenguaje de los datos, el sistema debe soportar por lo menos un lenguaje relacional que:

Tenga una sintaxis lineal.

Puede ser utilizado de manera interactiva.

Soporte operaciones de definición de datos, operaciones de manipulación de datos (actualización así como la recuperación), seguridad e integridad y operaciones de administración de transacciones.

– Regla 6: regla de actualización, todas las vistas que son teóricamente actualizables deben ser actualizables por el sistema.

– Regla 7: alto nivel de inserción, actualización, y cancelación, el sistema debe soportar suministrar datos en el mismo tiempo que se inserte, actualiza o esté borrando. Esto significa que los datos se pueden recuperar de una base de datos relacional en los sistemas construidos de datos de filas múltiples y/o de tablas múltiples.

– Regla 8: independencia física de los datos, los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuandoquiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

– Regla 9: independencia lógica de los datos, los cambios al nivel lógico (tablas, columnas, filas, etc.) no deben requerir un cambio a una solicitud basada en la estructura. La independencia de datos lógica es más difícil de lograr que la independencia física de datos.

– Regla 10: independencia de la integridad, las limitaciones de la integridad se deben especificar por separado de los programas de la aplicación y se almacenan en la base de datos. Debe ser posible cambiar esas limitaciones sin afectar innecesariamente las aplicaciones existentes.

– Regla 11: independencia de la distribución, la distribución de las porciones de la base de datos a las varias localizaciones debe ser invisible a los usuarios de la base de datos. Los usos existentes deben continuar funcionando con éxito:

Cuando una versión distribuida del SGBD se introdujo por primera vez

cuando se distribuyen los datos existentes se redistribuyen en todo el sistema.

– Regla 12: la regla de la no subversión, si el sistema proporciona una interfaz de bajo nivel de registro, a parte de una interfaz relacional, que esa interfaz de bajo nivel no se pueda utilizar para subvertir el sistema, por ejemplo: sin pasar por seguridad relacional o limitación de integridad. Esto es debido a que existen sistemas anteriormente no relacionales que añadieron una interfaz relacional, pero con la interfaz nativa existe la posibilidad de trabajar no relacionalmente.

## 7.- Recuperación y concurrencia.

## 8.- Administración.

# 9.- Lenguaje de Consultas.

## 1.- Sentencia Create Table.

CREATE TABLE table\_name (

column1 datatype,

column2 datatype,

column3 datatype,

....

);

ejemplo:

CREATE TABLE Persons (  
    id int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255),

Primary key(id)  
);

2.- Sentencia Alter Table.

## 3.- Sentencia Select.

Select [lista de campos | \*] from tabla;

Devuelve una tabla que contiene una lista de registros desde una tabla dada, ejemplos:

Suponga la tabla:

|  |  |
| --- | --- |
| Tabla: Costo | |
| Ciudad | Costo |
| Frontera | 50 |
| Sabancuy | 60 |
| Champoton | 100 |
| Frontera | 50 |
| Campeche | 145 |
| Puebla | 800 |
| DF | 1200 |
| Mérida | 250 |

A continuación, se dan ejemplos de consultas y su resultado:

Todos los datos de la tabla costo

|  |
| --- |
| Select \* from costo; |
| Ciudad Costo  ---------- --------  Frontera 50  Sabancuy 60  Champoton 100  Frontera 50  Campeche 145  Puebla 800  DF 1200  Mérida 250 |

Todos los nombres de las ciudades

|  |
| --- |
| Select ciudad from costo; |
| Ciudad  ----------  Frontera  Sabancuy  Champoton  Frontera  Campeche  Puebla  DF  Mérida |

Los datos de la tabla costo para las ciudades con costo mayor que 200

|  |
| --- |
| Select \* from costo where costo>200; |
| Ciudad Costo  ---------- --------  Puebla 800  DF 1200  Mérida 250 |

## 4.- sentencia insert.

Suponga que queremos insertar un registro en la tabla:

|  |  |
| --- | --- |
| Tabla: Alumnos | |
| Matricula | Nombre |
| 110967 | Juan Puc |
| 118967 | Rosa Chi |

Insert into alumnos (matricula, nombre) values (‘125634’,’Erika Uc’);

**Nota*:*** Debe tomar en cuenta que los valores de tipo texto como (varchar, char) deben ir entre apostrofes, los valores numéricos deben ir sin apostrofes, las fechas deben ir entre apóstrofes.

|  |  |
| --- | --- |
| Tabla: Alumnos | |
| Matricula | Nombre |
| 110967 | Juan Puc |
| 118967 | Rosa Chi |
| 125634 | Erika Uc |

## 5.- sentencia update.

Suponga que queremos modificar los datos de la Tuerca de ½ porque su peso debe ser 12 y su costo 15.

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla Partes | | | |
| clave | Parte | peso | costo |
| 110967 | Tornillo ¾ | 6 | 1 |
| 118967 | Tuerca ½ | 9 | 4 |
| 129087 | Arandela ¾ | 4 | 3 |

Update Partes set peso=12, costo=15 where clave=’118967’;

**Nota**: Debe tomar en cuenta que los valores de tipo texto como (varchar, char) deben ir entre apostrofes, los valores numéricos deben ir sin apostrofes, las fechas deben ir entre apóstrofes.

## 6.- Sentencia Like.

Suponga la tabla:

|  |  |
| --- | --- |
| Tabla: Alumnos | |
| Matricula | Nombre |
| 110967 | Juan Puc Alba |
| 118967 | Rosa Chi Cobos |
| 125634 | Erika Uc Paz |
| 122534 | Ross Díaz Montt |
| 102354 | Juan Ríos Perez |
| 114523 | José Juan Morales Rojas |

La siguiente sentencia muestra todos los registros cuyo nombre empiece con ‘Juan’;

|  |
| --- |
| Select \* from Alumnos where Nombre like ‘Juan%’; |
| Matricula Nombre  ------------ --------  110967 Juan Puc Alba  102354 Juan Ríos Perez |

Muestra todos los registros que contengan la palabra ‘Juan’ en el nombre:

|  |
| --- |
| Select \* from Alumnos where Nombre like ‘%Juan%’; |
| Matricula Nombre  ------------ --------  110967 Juan Puc Alba  102354 Juan Ríos Perez  114523 José Juan Morales Rojas |

## Objetivo de la práctica. 9.

Diseñar los movimientos de consultas, actualizaciones e inserción a la base de datos.

## Materiales de la práctica. 9.

* Computadora.
* Internet
* Suite de oficina

## Desarrollo de la práctica. 7.2.

Para la narrativa de proveedores y partes se forma un equipo de trabajo donde hay desarrolladores de la interfaz y fue diseñada ya la base de datos, usted es el encargado de diseñar las funciones PHP para hacer los movimientos a la base de datos, para lo que las funciones reciben datos en formato JSON los datos a insertar o actualizar por ejemplo y para las consultas devuelvan arreglos JSON con el resultado de las consultas. Apóyese en el material de introducción a PHP que proporcionó el profesor. Y las operaciones que se desean manejar son:

* Altas/ bajas/ modificaciones de partes.
* Altas/ bajas/ modificaciones de proveedores.
* Altas/ bajas/ modificaciones de tiposPartes.
* Altas/ bajas/ modificaciones de pedidos.
* Lista de partes.
* Lista de partes ordenada por tipo de parte junto con el nombre del tipo de la parte.
* Lista de partes con stock menor o igual al mínimo.
* Lista de pedidos de las partes para un tipo de parte dado.
* Lista de pedidos de las partes con un stock menor al mínimo.

## Entregable de la práctica. 7.2.

* Archivo pdf con el código php con sus respectivas sentencias sql.

# 10.- Administracion.

## 1. Metadatos

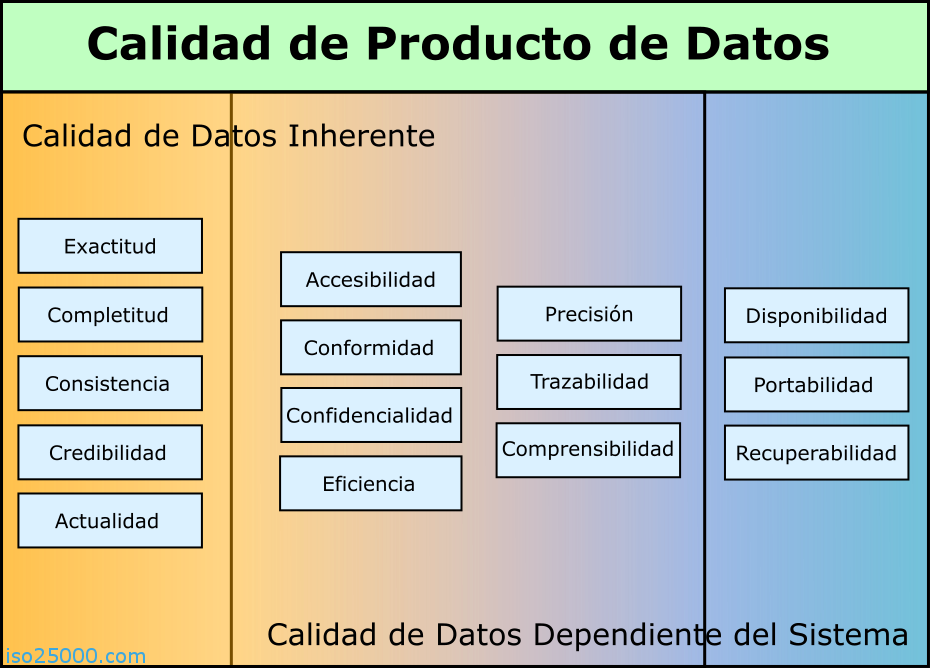
Los metadatos describen los datos, la información de la estructura de una tabla en un DBMS es un metadato, la tabla como conjunto de columnas y tuplas constituye los metadatos que describen el contenedor de los datos. Los metadatos también indican el tipo de datos que se almacenan, ahí se define si el campo edad es de tipo entero o numérico, si el campo nombre es de tipo varchar y cuál es su tamaño.

Estadar ISO: /IEC 25012

El modelo de Calidad de Datos representa los cimientos sobre los cuales se construye un sistema para la evaluación de un producto de datos. En un modelo de Calidad de Datos se establecen las características de Calidad de Datos que se deben tener en cuenta a la hora de evaluar las propiedades de un producto de datos determinado.

La Calidad del Producto de Datos se puede entender como el grado en que los datos satisfacen los requisitos definidos por la organización a la que pertenece el producto. Son precisamente estos requisitos los que se encuentran reflejados en el modelo de Calidad de Datos mediante sus características (Exactitud, Completitud, Consistencia, Credibilidad, Actualidad, Accesibilidad,).

El modelo de Calidad de Producto de Datos definido por el estándar ISO/IEC 25012 se encuentra compuesto por las 15 características que se muestran en la siguiente figura:



Las características de Calidad de Datos están clasificadas en dos grandes categorías:

**Calidad de Datos Inherente**: Se refiere al grado con el que las características de calidad de los datos tienen el potencial intrínseco para satisfacer las necesidades establecidas y necesarias cuando los datos son utilizados bajo condiciones específicas. Desde el punto de vista inherente, la Calidad de Datos se refiere a los mismos datos, en particular a:

Valores de dominios de datos y posibles restricciones (e.g., Reglas de Negocio gobernando la calidad requerida por las características en una aplicación dada).

Relaciones entre valores de datos (e.g., Consistencia).

Metadatos.

**Calidad de Datos Dependiente del Sistema**: Se refiere al grado con el que la Calidad de Datos es alcanzada y preservada a través de un sistema informático cuando los datos son utilizados bajo condiciones específicas.

Desde el punto de vista dependiente del sistema, la Calidad de Datos depende del dominio tecnológico en el que los datos se utilizan, y se alcanza mediante las capacidades de los componentes del sistema informático tales como: dispositivos hardware (e.g., Respaldo Software para alcanzar la Recuperabilidad), y otro software (e.g., Herramientas de migración para alcanzar la Portabilidad).

**Exactitud**

Grado en el que los datos representan correctamente el verdadero valor del atributo deseado de un concepto o evento en un contexto de uso específico.

Tiene dos principales aspectos:

Exactitud Sintáctica: cercanía de los valores de los datos a un conjunto de valores definidos en un dominio considerado sintácticamente correcto.

Exactitud Semántica: cercanía de los valores de los datos a un conjunto de valores definidos en un dominio considerado semánticamente correcto.

**Completitud**

Grado en el que los datos asociados con una entidad tienen valores para todos los atributos esperados e instancias de entidades relacionadas en un contexto de uso específico.

**Consistencia**

Grado en el que los datos están libres de contradicción y son coherentes con otros datos en un contexto de uso específico. Puede ser analizada en datos que se refieran tanto a una como a varias entidades comparables.

**Credibilidad**

Grado en el que los datos tienen atributos que se consideran ciertos y creíbles en un contexto de uso específico. La credibilidad incluye el concepto de autenticidad (la veracidad de los orígenes de datos, atribuciones, compromisos).

**Actualidad**

Grado en el que los datos tienen atributos que tienen la edad correcta en un contexto de uso específico.

**Calidad de Datos Inherente y Dependiente del Sistema**

**Accesibilidad**

Grado en el que los datos pueden ser accedidos en un contexto específico, particularmente por personas que necesiten tecnologías de apoyo o una configuración especial por algún tipo de discapacidad.

**Conformidad**

Grado en el que los datos tienen atributos que se adhieren a estándares, convenciones o normativas vigentes y reglas similares referentes a la calidad de datos en un contexto de uso específico.

**Confidencialidad**

Grado en el que los datos tienen atributos que aseguran que los datos son sólo accedidos e interpretados por usuarios autorizados en un contexto de uso específico. La confidencialidad es un aspecto de la seguridad de la información (junto con la disponibilidad y la integridad) definida como en ISO/IEC 13335-1:2004.

**Eficiencia**

Grado en el que los datos tienen atributos que pueden ser procesados y proporcionados con los niveles de rendimiento esperados mediante el uso de cantidades y tipos adecuados de recursos en un contexto de uso específico.

**Precisión**

Grado en el que los datos tienen atributos que son exactos o proporcionan discernimiento en un contexto de uso específico.

**Trazabilidad**

Grado en el que los datos tienen atributos que proporcionan un camino de acceso auditado a los datos o cualquier otro cambio realizado sobre los datos en un contexto de uso específico.

**Comprensibilidad**

Grado en el que los datos tienen atributos que permiten ser leídos e interpretados por los usuarios y son expresados utilizando lenguajes, símbolos y unidades apropiados en un contexto de uso específico. Cierta información sobre la comprensibilidad puede ser expresada mediante metadatos.

**Calidad de Datos Dependiente del Sistema**

**Disponibilidad**

Grado en el que los datos tienen atributos que permiten ser obtenidos por usuarios y/o aplicaciones autorizados en un contexto de uso específico.

**Portabilidad**

Grado en el que los datos tienen atributos que les permiten ser instalados, reemplazados o eliminados de un sistema a otro, preservando el nivel de calidad en un contexto de uso específico.

**Recuperabilidad**

Grado en el que los datos tienen atributos que permiten mantener y preservar un nivel específico de operaciones y calidad, incluso en caso de fallos, en un contexto de uso específico.

## 2 Seguridad en bases de datos.

Estudie el contenido de: <https://www.ibm.com/mx-es/cloud/learn/database-security>

Haga una presentación sobre este contenido.

## 3 Seguridad en bases de datos.

https://www.adictosaltrabajo.com/2009/12/08/mysql-replicacion/