

BASE DE DATOS



CARRERA: ANALISIS DE SISTEMAS
SEMESTRE: TERCERO

INTRODUCCIÓN A LA ASIGNATURA

En el presente eBook, se abordará acerca de las Bases de Datos y el papel fundamental en el campo de la informática y la ingeniería de software. Su objetivo principal es enseñar a los estudiantes cómo almacenar, gestionar y tratar datos de manera eficiente y efectiva. A medida que las organizaciones generan y utilizan grandes volúmenes de información, la capacidad de manejar bases de datos se vuelve cada vez más crucial.

Temas Clave que se Abordan en la Asignatura:

1. Conceptos Básicos: Se introducen los conceptos fundamentales de las bases de datos, incluyendo qué son, por qué son importantes y cómo se utilizan en el mundo real.

2. Modelos de Datos: Se estudian diferentes modelos de datos, como el modelo relacional, el modelo orientado a objetos y los modelos no relacionales (NoSQL). Cada uno tiene sus propias características y casos de uso.

3. Lenguaje SQL: Se enseña SQL (Structured Query Language), que es el lenguaje estándar para interactuar con bases de datos relacionales. Los estudiantes aprenden a realizar consultas, insertar, actualizar y eliminar datos.

4. Diseño de Bases de Datos: Se abordan las mejores prácticas para diseñar bases de datos eficientes y escalables, incluyendo la normalización y la creación de esquemas.

5. Gestión de Bases de Datos: Se exploran los sistemas de gestión de bases de datos (DBMS), que son herramientas que permiten a los usuarios crear y gestionar bases de datos. Se analizan diferentes DBMS populares como MySQL, PostgreSQL y MongoDB.

6. Seguridad y Control de Acceso: Se discuten las medidas necesarias para proteger los datos y garantizar que solo los usuarios autorizados puedan acceder a la información.

7. Transacciones y Concurrencia: Se estudian los conceptos de transacciones en bases de datos y cómo manejar múltiples usuarios que acceden a la base de datos simultáneamente.

Importancia de la Asignatura:

La asignatura de Bases de Datos es esencial para cualquier profesional que desee trabajar en el ámbito de la tecnología, ya que la mayoría de las aplicaciones y sistemas dependen de bases de datos para funcionar.

Además, el conocimiento en este campo abre muchas oportunidades laborales en áreas como desarrollo de software, análisis de datos, administración de sistemas y más encontrará definiciones teóricas referente a las base de datos, se desarrollará su usos, objetivos, ventajas, para qué sirve, diseño, modelo conceptual, modelo Relacional, entre otros aspectos inherentes.

Adicional, se profundizará sobre los tipos de bases de datos existen, y algunos ejemplos prácticos, donde podrá adquirir los conocimientos necesarios que le permitan modelar, diseñar, normalizar y consultar vía SQL (Lenguaje Estructurado de Consulta) bases de datos relacionales.

NICOLÁS A. RÍOS M

INGENIERO EN SISTEMAS

ORCID iD: 0009-0004-0813-9872

Una publicación de:



PROGRAMA

Contenido.

- ✓ Bases de Datos, Reseña Histórica, Concepto y Ventajas.
- ✓ Concepto de independencia física y lógica de los datos, Arquitectura de las Bases de Datos.
- ✓ Entidad, Relación, Concepto, Atributos en un MER.
- ✓ Diagramas de Entidad / Relación, Modelos de Entidad / Relación, Estructura del Modelo relacional.
- ✓ Modelo relacional, Conceptos básicos de dominio, atributo, relación y claves.
- ✓ Conversión de un diagrama E/R a un modelo relacional.
- ✓ Teoría de normalización, Definición, Objetivos.
- ✓ Dependencias funcionales entre atributos, Definición Primera, segunda y tercera forma normal, Definición, Identificación de relaciones.
- ✓ Operadores fundamentales del álgebra relacional.
- ✓ Sentencias básicas del Lenguaje relacional SQL (SELECT, INSERT, UPDATE, CRÉATE TABLE)

INDICE

| | |
|--|----|
| UNIDAD I / BASE DE DATOS | 10 |
| ❖ Campo: | 10 |
| ❖ Clave | 10 |
| ❖ Consulta:..... | 10 |
| ❖ DBMS:..... | 10 |
| ❖ Registro: | 10 |
| ❖ JDBC:..... | 11 |
| ❖ Relación | 11 |
| ❖ Tabla | 11 |
| ❖ Tupla..... | 11 |
| ❖ Dato | 11 |
| ❖ ODBC | 11 |
| ❖ Clave Foránea | 11 |
| ❖ Cláusula | 12 |
| ❖ SGBD..... | 12 |
| Bases de datos analíticas..... | 12 |
| Bases de datos dinámicas..... | 13 |
| Sistema de Gestión de la Base de Datos (SGBD)..... | 14 |
| Modelos de bases de datos..... | 15 |
| Bases de datos jerárquicas..... | 16 |
| Bases de datos de red | 16 |
| Bases de datos relacionales | 17 |
| Características de las BD. | 17 |
| Ventajas de las BD | 18 |
| Manejadores de BD..... | 18 |
| ✓ MySql..... | 18 |

| | | |
|----|--|----|
| ✓ | PostgreSql y Oracle..... | 19 |
| ✓ | Access..... | 19 |
| ✓ | Microsoft SQL Server..... | 19 |
| | Independencia Física y Lógica de los Datos..... | 19 |
| ✓ | Independencia Física de datos | 19 |
| ✓ | Independencia Lógica de datos..... | 20 |
| | Arquitectura Tier 1 | 21 |
| | Arquitectura Tier 2 | 21 |
| | Arquitectura Tier 3 | 21 |
| | Niveles de una Arquitectura de Bases de datos..... | 22 |
| 1. | Nivel Interno (Físico): | 22 |
| 2. | Nivel Conceptual (Lógico):..... | 22 |
| 3. | Nivel Externo (de Usuario o de Vista): | 22 |
| | RECURSOS INTERACTIVOS..... | 24 |
| | Actividades Sugeridas..... | 24 |
| | Actividad 1..... | 24 |
| | Actividad 2..... | 24 |
| | UNIDAD II / DISEÑO DE BASE DE DATOS..... | 25 |
| | Modelos de los datos. | 25 |
| | Modelo entidad-relación..... | 25 |
| ○ | Rectángulos | 26 |
| ○ | Elipses..... | 26 |
| ○ | Rombos..... | 26 |
| ○ | Líneas..... | 26 |
| | Modelo relacional | 27 |
| | RECURSOS INTERACTIVOS..... | 29 |
| | Actividades Sugeridas..... | 29 |

| | |
|--|----|
| Actividad 3..... | 29 |
| Actividad 4..... | 29 |
| UNIDAD III / PROCESO DE NORMALIZACIÓN..... | 30 |
| Normalización..... | 30 |
| Primera forma normal (1nf)..... | 30 |
| Segunda forma normal (2nf)..... | 31 |
| Tercera forma normal (3nf)..... | 32 |
| RECURSOS INTERACTIVOS..... | 34 |
| Actividades Sugeridas..... | 34 |
| Actividad 5..... | 34 |
| UNIDAD IV / OPERADORES FUNDAMENTALES DEL ÁLGEBRA RELACIONAL..... | 35 |
| Lenguaje SQL..... | 35 |
| ▪ UNION:..... | 36 |
| ▪ INNER JOIN:..... | 36 |
| ▪ LEFT JOIN:..... | 37 |
| ▪ RIGHT JOIN:..... | 37 |
| Consultas básicas en lenguaje estructurado..... | 38 |
| FROM..... | 38 |
| WHERE:..... | 38 |
| GROUP BY..... | 38 |
| HAVING..... | 38 |
| ORDER BY..... | 38 |
| La sintaxis de una consulta básica..... | 38 |
| Otra consulta aplicando una cláusula..... | 39 |
| Operadores Aritméticos..... | 39 |
| Operadores Lógicos..... | 40 |
| Operadores Relacionales..... | 40 |

| | |
|---|----|
| Funciones de agregado..... | 41 |
| RECURSOS INTERACTIVOS..... | 43 |
| Actividades Sugeridas..... | 43 |
| Actividad 6..... | 43 |
| Actividad 7..... | 43 |
| REFERENCIAS CONSULTADAS..... | 44 |
| ANEXO A..... | 46 |
| AULA CLIC. CURSOS DE INFORMÁTICA GRATUITOS..... | 47 |
| Guía Tipos de Datos 1/6..... | 47 |
| Guía Tipos de Datos 2/6..... | 48 |
| Guía Tipos de Datos 3/6..... | 49 |
| Guía Tipos de Datos 4/6..... | 50 |
| Guía Tipos de Datos 5/6..... | 51 |
| Guía Tipos de Datos 6/6..... | 52 |
| ANEXO B..... | 53 |
| EJERCICIOS DE BASES DE DATOS. SOLUCIONES..... | 54 |
| Ejercicio 1..... | 55 |
| Ejercicio 2..... | 56 |
| Ejercicio 3..... | 57 |
| Ejercicio 4..... | 58 |
| Ejercicio 5..... | 59 |
| Ejercicio 6..... | 60 |
| Ejercicio 7..... | 61 |
| Ejercicio 8..... | 62 |
| Ejercicio 9..... | 63 |
| Ejercicio 10..... | 64 |
| Ejercicio 11..... | 65 |

| | |
|--|----|
| Ejercicio 12 | 66 |
| Ejercicio 13 | 67 |
| Ejercicio 14 | 68 |
| Ejercicio 15 | 69 |
| Ejercicio 16 | 70 |
| Ejercicio 17 | 71 |
| Ejercicio 18 | 72 |
| Ejercicio 19 | 73 |
| Ejercicio 20 | 74 |
| ANEXO C | 75 |
| LA CONECTIVIDAD ABIERTA DE BASE DE DATOS | 76 |
| (ODBC) | 76 |
| ODBC | 76 |
| ANEXO D | 78 |
| PRÁCTICA DE CONSULTAS SQL | 79 |
| Caso de Estudio 1 | 79 |
| Caso de Estudio 2 | 79 |
| Caso de Estudio 3 | 79 |
| Caso de Estudio 4 | 80 |
| Caso de Estudio 5 | 80 |
| Caso de Estudio 6 | 80 |
| Caso de Estudio 7 | 80 |

UNIDAD I / BASE DE DATOS

De acuerdo a sus siglas del español BD (*Base de Datos*) o en inglés DB (*Data Base*). Lo podemos definir una BD cómo, aquella entidad utilizada para almacenar datos de manera estructura.

Para dar introducción al contenido de la unidad, es necesario reforzar algunos términos que serán empleados a lo largo del contenido previsto.

- ❖ Campo: es la unidad básica de una base de datos. Un campo puede ser, por ejemplo, el nombre de una persona. Los nombres de los campos no pueden empezar con espacios en blanco y caracteres especiales. No pueden llevar puntos, ni signos de exclamación o corchetes.
- ❖ Clave principal: identificado como “CP” ó “PK”, usada en las BD para identificar de forma unívoca (registro único) cada registro de la tabla. Por ejemplo, en una tabla de alumnos podría ser su número código.
- ❖ Consulta: mediante las consultas tendrás la posibilidad de obtener toda la información contenida en las tablas añadiendo interesantes funcionalidades.
- ❖ DBMS: conjunto de programas destinados a manejar la creación y todos los accesos a las bases de datos.
- ❖ Registro: es el conjunto de información referida a una misma unidad.

- ❖ **JDBC:Conectividad de BD Java** (Java DatabaseConnectivity, JDBC) es una especificación de la interfaz de aplicación de programa (applicationprogram interface, API) para conectar los programas escritos en Java a los datos en bases de datos populares.
- ❖ **Relación:** permite establecer vínculos o relaciones entre las tablas que lo componen. El objetivo de estas relaciones sería principalmente evitar la duplicidad de información y en consecuencia, optimizar el rendimiento de la base de datos.
- ❖ **Tabla:** unidad donde crearemos el conjunto de datos de nuestra base de datos. Estos datos estarán ordenados en columnas verticales. En ella se definen los campos y sus características.
- ❖ **Tupla:** también se denomina de este modo a un registro o fila de una tabla.
- ❖ **Dato:** es la unidad básica de información que se almacena y se organiza dentro de un campo o espacio dentro de un sistema, que a su vez representa un valor concreto (numero, letras, fecha, otros).
- ❖ **ODBC:Open DatabaseConnectivity.** Es un protocolo y una interfaz de programación de aplicaciones (API) de estándar abierto que permite a las aplicaciones acceder a los datos de cualquier base de datos en la que haya un controlador ODBC.
- ❖ **Clave Foránea:** Conocida como llave foránea (Foreign Key, FK), es una columna o conjunto de columnas que se relaciona a la clave primaria de otra tabla. Las claves foráneas se utilizan para vincular dos tablas y establecer una relación entre ellas.

- ❖ Cláusula: son las condiciones que alteran el resultado de nuestras consultas a la base de datos.
- ❖ SGBD:(Sistema Gestor de Base de Datos) son todas la aplicaciones que permiten la administración o gestión de la Base de Datos..

Las Bases de Datos, de acuerdo a su función éstas pueden dividirse en dos grupos, considerando su funcionamiento primordial:

Bases de datos analíticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

Figura 1.

Analítica de Datos



Fuente: <https://www.inesdi.com/blog/analitica-datos-ciberseguridad/>

Bases de datos dinámicas

Éstas son bases de datos más dinámicas, orientadas a almacenar información que es modificada con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consultas.

Figura 2.

Beneficios de armar una base de datos en Microsoft Excel



Fuente: <https://www.ninjaexcel.com/excel-en-el-trabajo/base-de-datos-microsoft-excel/>

Una BD se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios.

Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos.

Esta descripción es lo que se denomina *metadatos*, se almacena en el *diccionario de datos* o *catálogo* y es lo que permite que exista independencia de datos lógica-física.

Antes de existir las bases de datos se trabajaba con sistemas de ficheros. Los sistemas de ficheros surgieron al informatizar el manejo de los archivadores manuales para proporcionar un acceso más eficiente a los datos almacenados en los mismos.

Un sistema de ficheros sigue un modelo descentralizado, en el que cada departamento de la empresa almacena y gestiona sus propios datos mediante una serie de programas de aplicación escritos especialmente para él.

Sistema de Gestión de la Base de Datos (SGBD)

Es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma.

Se denomina sistema de bases de datos al conjunto formado por la base de datos, el SGBD y los programas de aplicación que dan servicio a la empresa u organización.

Generalmente, un SGBD proporciona los servicios que se citan a continuación:

- El SGBD permite la definición de la base de datos mediante un lenguaje de definición de datos. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos.
- El SGBD permite la inserción, actualización, eliminación y consulta de datos mediante un lenguaje de manejo de datos.
- El SGBD proporciona un acceso controlado a la base de datos mediante:
 - Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos.

- Un sistema de integridad que mantiene la integridad y la consistencia de los datos.
- Un sistema de control de concurrencia que permite el acceso compartido a la base de datos.
- Un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del hardware o del software.
- Un diccionario de datos o catálogo, accesible por el usuario, que contiene la descripción de los datos de la base de datos.

Modelos de bases de datos

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores.

Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

Bases de datos jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos.

El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

Bases de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que TIENE una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Bases de datos relacionales

Éste es el modelo más utilizado en la actualidad, basado en relaciones. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas".

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos.

La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. A

Características de las BD.

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.

- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

Ventajas de las BD

- Independencia de los datos frente a los tratamientos y viceversa.
- Coherencia de los resultados.
- Mejor disponibilidad de los datos para el conjunto de los usuarios.
- Mayor valor informativo.
- Mejor y más normalizada documentación de la información, la cual está integrada con los datos.
- Reducción del espacio de almacenamiento.
- La seguridad de los datos.

Manejadores de BD

- ✓ MySQL: es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.

- ✓ PostgreSQL y Oracle: Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.
- ✓ Access: Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa access, el cual crea un archivo .mdb con la estructura ya explicada.
- ✓ Microsoft SQL Server: es una base de datos más potente que access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones..

Independencia Física y Lógica de los Datos

Para entender sobre la independencia Física y Lógica de los Datos es necesario precisar que la capacidad de poder modificar una definición de esquema en un nivel sin que vea afectado una definición de esquema en el siguiente nivel más alto se llama Independencia de datos.

Existe dos niveles de independencia de datos:

- ✓ Independencia Física de datos: Es la capacidad para modificar el esquema físico sin provocar que los programas de aplicación tengan que reescribirse.

Las modificaciones en el nivel físico son ocasionalmente necesarias para mejorar el funcionamiento.

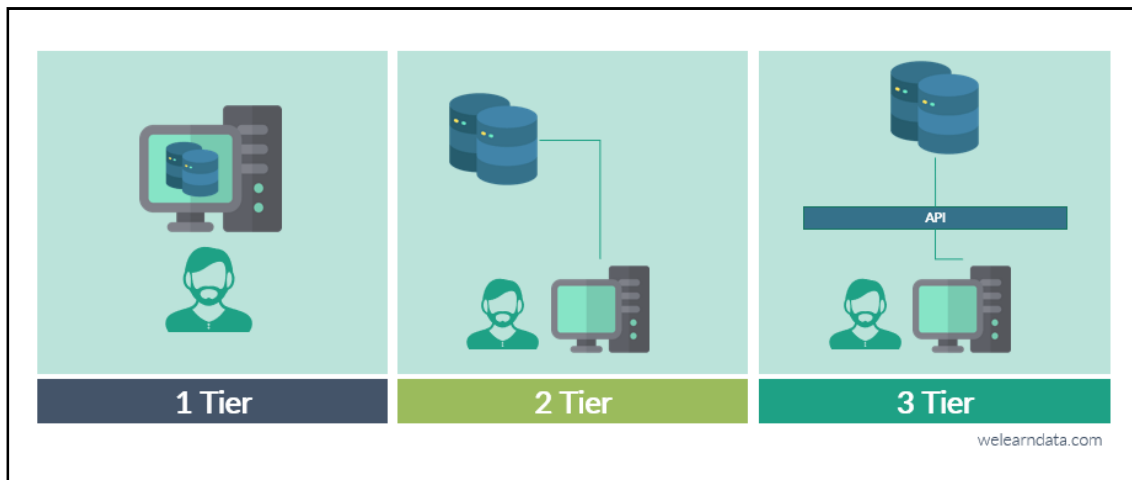
- ✓ Independencia Lógica de datos: Es la capacidad para modificar el esquema lógico sin causar que los programas de aplicación tengan que rescribirse.

Las modificaciones en el nivel lógico son necesarias siempre que la estructura lógica de la base de datos se altere.

La independencia de datos lógica es más fácil de proporcionar que la independencia de datos física, ya que los programas de aplicación son fuertemente dependientes de la estructura lógica de los datos a los que ellos acceden.

Figura 3.

Tipos de arquitecturas de bases de datos



Fuente: <https://welearndata.com/bases-de-datos/arquitectura-base-datos/>

Arquitectura Tier 1

Se trata de una arquitectura donde servidor, cliente y base de datos residen en la misma máquina. Aquí se alojarán la aplicación, los datos, el DBMS y la interfaz, todo en el mismo sitio. Puede resultar útil para entornos de pruebas, pero no es lo habitual en producción.

Arquitectura Tier 2

La típica aplicación cliente-servidor. La capa de presentación de datos se encuentra en otra máquina o máquinas, independiente del servidor donde reside la base de datos, llamado tier 2.

De esta forma, la base de datos no está expuesta a los servicios de los clientes y podemos tener varios usuarios conectados a la base de datos.

Arquitectura Tier 3

Es la más común. Separamos la capa de presentación (los usuarios), la capa de aplicación, que hace de intermediador comunicando las peticiones de los usuarios con la base de datos, y la capa con la base de datos en otro servicio donde estarían definidas las relaciones entre las tablas de las bases de datos.

Cabe destacar que en la capa de aplicación es donde se define la lógica de comunicación entre usuario y BD, y por consiguiente abstrae al usuario de la

comunicación con la BD. Es con esta pieza de software (middleware) con quien se comunica, y no con la BD.

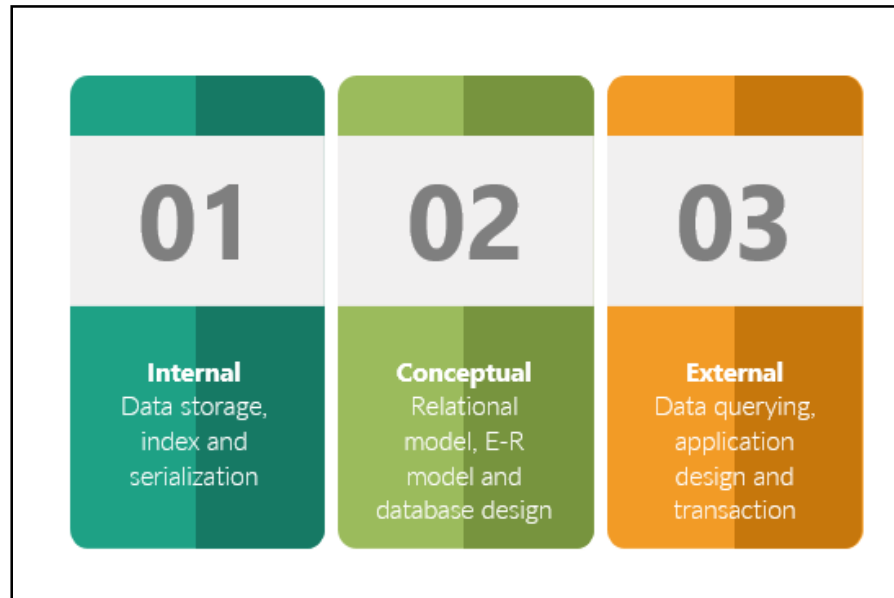
Niveles de una Arquitectura de Bases de datos.

Las bases de datos tienen una arquitectura a tres niveles compuesta por:

1. Nivel Interno (Físico): Este nivel se refiere a cómo los datos se almacenan físicamente en el dispositivo de almacenamiento, como discos duros. Incluye detalles sobre la estructura de almacenamiento, la organización de datos en bloques, la gestión del espacio y la optimización del rendimiento.
2. Nivel Conceptual (Lógico): En este nivel, se define la estructura lógica de la base de datos, independientemente de cómo se almacenan físicamente los datos. Se establecen las entidades, relaciones, restricciones y reglas de integridad. Es un nivel de abstracción que describe la base de datos en su conjunto.
3. Nivel Externo (de Usuario o de Vista): se crean vistas personalizadas de los datos para usuarios finales o aplicaciones específicas. Cada vista muestra solo la porción de la base de datos que es relevante para el usuario o la aplicación, proporcionando una capa de abstracción y seguridad.

Figura 5.

Niveles de una arquitectura de base de datos: interno, conceptual y externo.



Fuente: <https://welearndata.com/bases-de-datos/arquitectura-base-datos>

RECURSOS INTERACTIVOS

A continuación se anexa el siguiente link que contiene un recurso audio visual para reforzar ésta primera parte de la unidad de estudio.

Link: <https://youtu.be/6S8A-1jBD5Y>

Actividades Sugeridas.

Actividad 1. Informe Analítico (Investigar)

- Reseña Histórica de las Bases de Datos.
- Visión General de una BD.
- Objetivos de las BD.
- Algún punto que considere pertinente (desarrolle).

Portada Identificación. Contenido. Referencias Bibliográficas consultadas.

Actividad 2. Presentación (Digital, Hoja papel bond) Grupal

Por cada Manejador / Gestor de BD

- Breve Reseña, origen y año, identificar si es de baja, media o alta gama.
- Bajo que Infraestructura trabaja (SO base).
- Aplicaciones importantes en el mercado existente que la usan.
- Agregar imágenes del entorno de trabajo.
- Algún dato de interés que considere relevante.

UNIDAD II / DISEÑO DE BASE DE DATOS

Modelos de los datos.

En la estructura de la base de datos se encuentra el modelo de datos, simplemente es una colección de herramientas conceptuales para hacer una descripción de los datos, las relaciones, la semántica y las restricciones de consistencia.

Por lo tanto el concepto de un modelo de datos, lo podemos describir en dos modelos de datos: el modelo entidad relación y el modelo relacional.

Modelo entidad-relación.

El modelo de datos entidad-relación (ER) está compuesto de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es una “Cosa” u “Objeto” que puede distinguirse de otros objetos.

Por ejemplo, cada ESTUDIANTE es una entidad, y las MATERIAS pueden ser consideradas entidades.

Las entidades se describen en una base de datos mediante un conjunto de atributos.

De acuerdo a esto, presentamos otro ejemplo, los atributos id_materia y nombre_mat describen una materia particular de un instituto y pueden ser

atributos del conjunto de entidades MATERIAS. De igual forma, los atributos nombre, dirección, ciudad pueden describir una entidad ESTUDIANTE.

Un atributo extra, código, se usa para identificar a los Estudiantes (puede ser posible que exista dos estudiantes con el mismo nombre, dirección y ciudad. Por lo tanto se debe asignar un identificador único a cada uno.

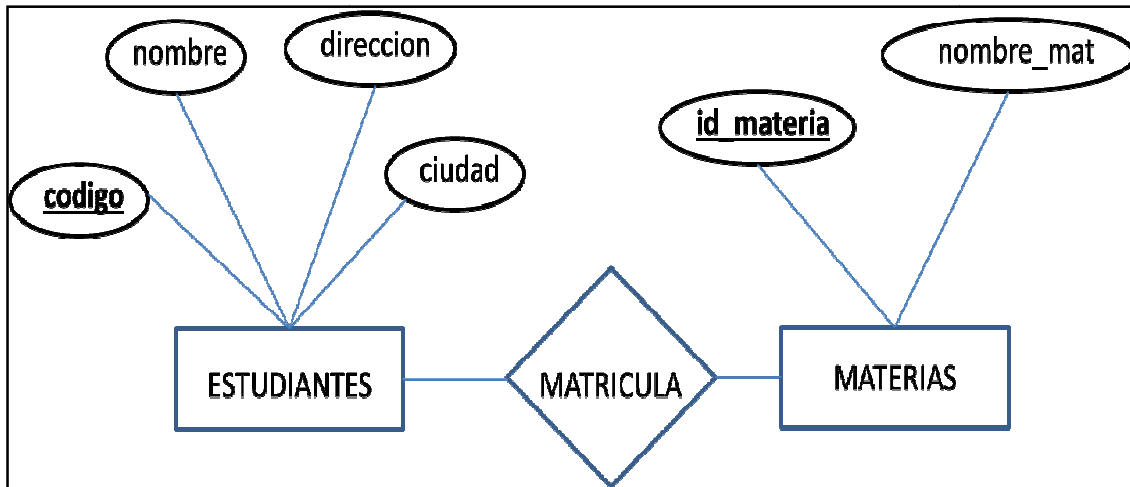
Una relación, es una asociación entre varias entidades. Por ejemplo, una relación matrícula asocia un estudiante con cada materia que tiene.

El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente conjunto de entidades y conjunto de relaciones.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama ER, que consta de los siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombos, que representan relaciones entre conjuntos de entidades.
- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

Figura 6.
Modelo Entidad Relación



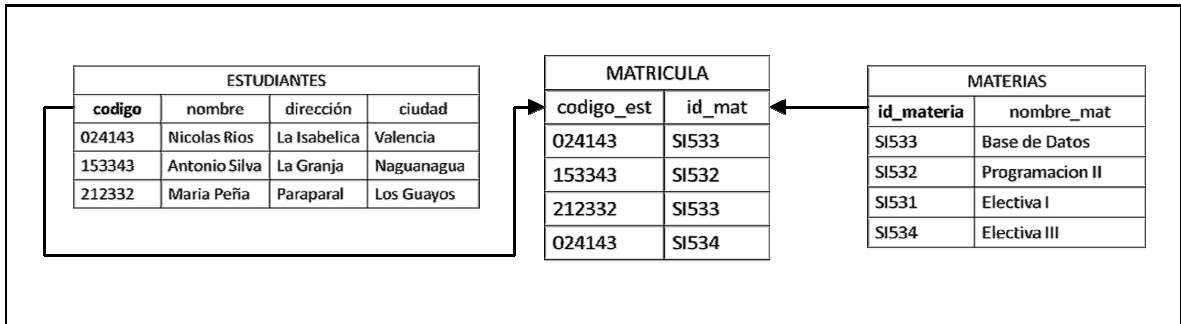
Fuente: Ríos Nicolas, 2024.

Cada componente se etiqueta con la entidad o relación que representa. Como ilustración, considérese parte de una base de datos de un sistema inscripción de un Instituto Educativo que contiene Estudiantes y Materias de la institución.

Modelo relacional

En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único. En la Figura 7 se presenta ejemplo de base de datos relacional contenida las tablas anteriormente trabajadas: la primera muestra los *ESTUDIANTES*, la segunda, las *MATERIAS*, y la tercera, las *MATRICULAS* que pertenecen a cada estudiantes.

Figura 7.
Modelo Relacional



Fuente: Rios Nicolas, 2024

RECURSOS INTERACTIVOS

De acuerdo al contenido suministrado en la Unidad II, realizar las siguientes actividades.

Actividades Sugeridas.

Actividad 3. Diseño de una Base Datos (propia)

- Realizar modo vista de tablas, identificar (campo principal)
- Realizar Diagrama Entidad Relación (ER).
- Realizar Diagrama Relacional.
- Breve descripción de la BD.

Recursos, Guía Tipos de Datos. (Ver Anexo A)

Actividad 4. Resolver Ejercicios Prácticos de ER.

Se Adjunta una guía de problemas propuestos para ser desarrollados. (Ver Anexo B)

Partiendo desde el Diagrama ER, realizar el Diagrama Relacional con sus respectivas tablas y relaciones.

UNIDAD III / PROCESO DE NORMALIZACIÓN.

Normalización.

Normalizar, es el proceso de simplificación de los datos ingresado en una BD.

Para que Normalizar?

- Para almacenar mayor cantidad de datos con menor espacio.
- Eliminar redundancia.
- Ordenar los Datos

Una forma normal es una regla sobre las dependencias permisibles. Cada forma normal elimina cierto tipo de redundancias.

Cada forma normal sucesiva refina la forma normal previa para eliminar otro tipo de redundancia adicional.

Primera forma normal (1nf)

La 1nf prohíbe la anidación o repetición de grupos en las tablas. Una tabla que no esté en 1nf está desnormalizada o sin normalizar.

Figura 8
Tabla sin Normalizar

| TABLA1 | | | | | |
|--------|---------|-----------|----------------|---------------|----------------------|
| Código | Nombre | Apellido | Asignatura | Id_asignatura | Carrera |
| 024143 | Nicolas | Rios | Base de Datos | SI533 | Análisis de Sistemas |
| 024143 | Nicolas | Rios | Electiva 3 Red | SI534 | Análisis de Sistemas |
| 024143 | Nicolas | Rios | Programacion | SI532 | Análisis de Sistemas |
| 212223 | Juan | Hernandez | Base de Datos | SI533 | Análisis de Sistemas |

Fuente: RiosNicolas, 2024

La 1nf, se basa en identificar los Datos repetidos. De acuerdo al ejemplo TABLA1 (figura8), son Codigo, Nombre, Apellido, Carrera... los extraemos como Tabla ALUMNO, aplicando la 1nf el resultado se muestra en la siguiente figura 9.

Figura 9.
Tabla con la 1fn aplicada

| ALUMNO | | | |
|-----------------|---------------|-----------------|----------------------|
| * Codigo | Nombre | Apellido | Carrera |
| 024143 | Nicolas | Rios | Análisis de Sistemas |
| 212223 | Juan | Hernandez | Análisis de Sistemas |

| MATERIA (1nf) | | |
|----------------------|-------------------|----------------------|
| Codigo_alumno | Asignatura | Id_asignatura |
| 024143 | Base de Datos | SI533 |
| 024143 | Electiva 3 Red | SI534 |
| 024143 | Programacion | SI532 |
| 212223 | Base de Datos | SI533 |

Fuente: RiosNicolas, 2024.

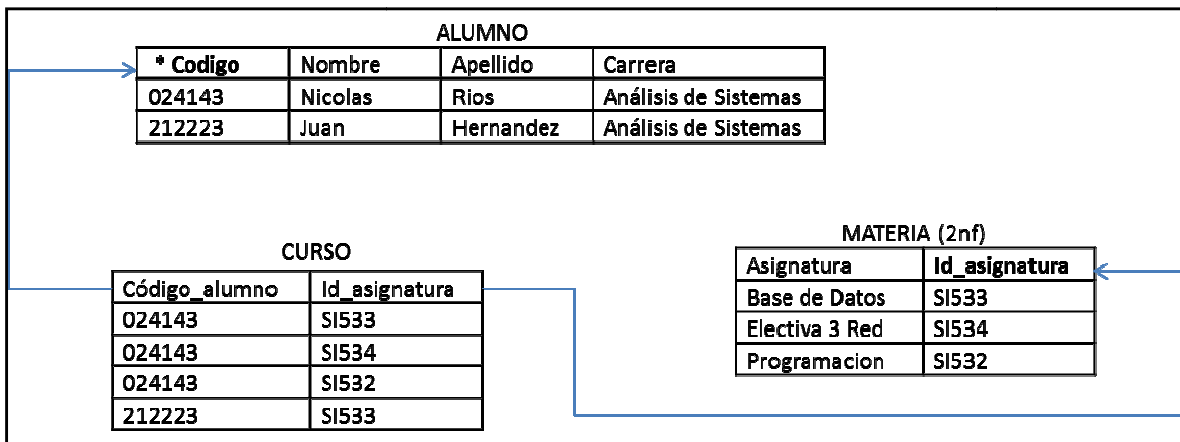
Los campos restantes quedarían, tabla MATERIA, nótese que se mantiene la estructura de Datos. No se ha perdido información alguna.

Segunda forma normal (2nf)

En la 2nf, para proceder, la tabla debe estar en 1nf y se debe identificar las dependencias funcionales o transitivas.

De acuerdo a lo obtenido arriba (figura 9), se observa que MATERIA repiten datos asignatura: *Base de Datos*, podemos aplicar la 2nf, quedando de la siguiente forma, ver figura 10.

Figura 10.
Tabla con la 2nf aplicada.



Fuente: RiosNicolas, 2024.

Ahora, se evidencia que se ha minimizado la tabla materia, no existe datos redundante.

Se ha creado una tabla Nueva para mantener la Relación de Cursos - Materias - Alumnos.

Tercera forma normal (3nf)

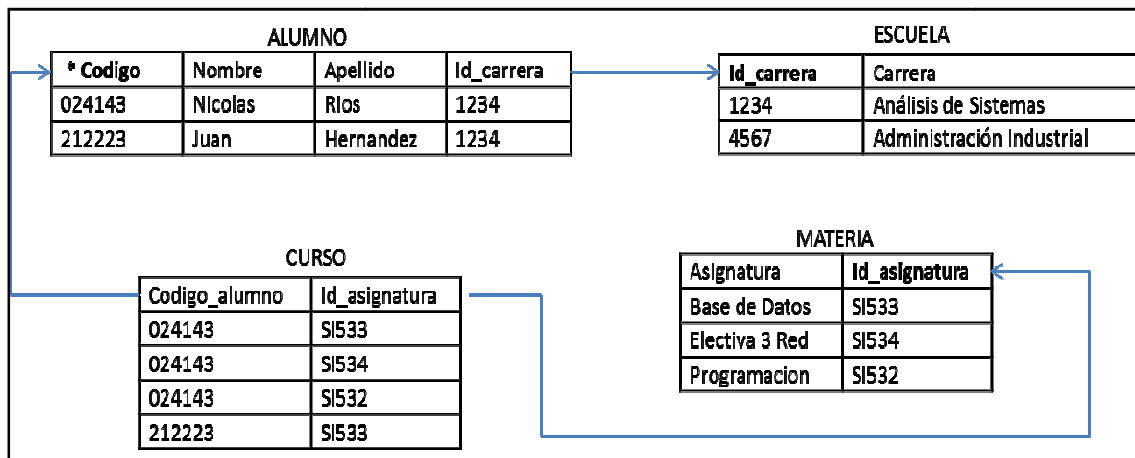
La 3nf, para proseguir con esta forma es preciso que la tabla esté en la 2nf y no debe tener dependencias funcionales o transitivas.

Para este punto se observa que solo repiten datos en la tabla ALUMNO en el campo carrera (no tiene dependencias funcionales o transitivas), entonces procedemos a crear una tabla nueva para agregar las escuelas – carreras.

Tomaremos la ilustración de la figura 10, aplicamos la 3nf para obtener las siguientes tablas, ver figura 11.

Figura 11.

Tabla con la 3fn aplicada.



Fuente: RiosNicolas, 2024.

Verificando el resultado final, observe que se mantiene la estructura de datos, bajo una entidad relación, optimizada sin pérdida de datos.

RECURSOS INTERACTIVOS.

A continuación se anexa el siguiente link que contiene un recurso audio visual para reforzar ésta primera parte de la unidad de estudio.

1. Link: <https://youtu.be/m7kpSO6kqY8>
2. Link: <https://youtu.be/MdYvCMbQji4?list=LL>

Actividades Sugeridas.

Actividad 5. Aplicar proceso de Normalización.

- De la actividad 3 realizada Diseño de una BD propia.

Aplicar el principio de normalización. Paso a Paso.

Portada Identificación.

Contenido del desarrollo de las formas.

UNIDAD IV / OPERADORES FUNDAMENTALES DEL ÁLGEBRA RELACIONAL.

Son el conjunto de operadores para manipular las bases de datos relacionales. Cada operador usa una o dos tablas como entrada y genera una nueva tabla como salida.

Lenguaje SQL.

StructuredQueryLanguage - Lenguaje de Consulta Estructurada. Es utilizado para consultar las tablas de BD. Los mismos se dividen en Sub-lenguajes, ver la siguiente figura.

Figura 12.
Sub-lenguaje SQL

| Sublenguaje SQL | Comandos | Ejemplos |
|------------------------------------|---|--------------------------|
| Data Definition Language (DDL) | Comandos para definir el esquema de la base de datos: crear, modificar y eliminar tablas de la base de datos; definir claves primarias, claves externas y limitaciones (constraints). | CREATE TABLE, DROP TABLE |
| Data Manipulation Language (DML) | Comandos de manipulación de datos: modificar, insertar y eliminar registros. | INSERT, UPDATE |
| Data Query Language (DQL) | Comandos para consultar y acondicionar los datos. | SELECT |
| Data Control Language (DCL) | Comandos para la gestión de derechos. | GRANT, REVOKE |
| Transaction Control Language (TCL) | Comandos para el control de las transacciones. | COMMIT, ROLLBACK |

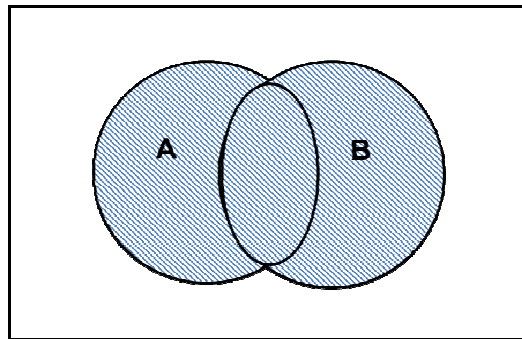
Fuente: RiosNicolas, 2024

Las operaciones con conjuntos también conocidas como álgebra de conjuntos, nos permiten realizar operaciones sobre los éstos para obtener otro conjunto.

- **UNION:** Combina dos conjuntos de resultados; los conjuntos de resultados deben tener columnas del mismo tipo en el mismo orden. Sus filas están unidas.

Figura 13.

Expresión de Conjunto UNION.

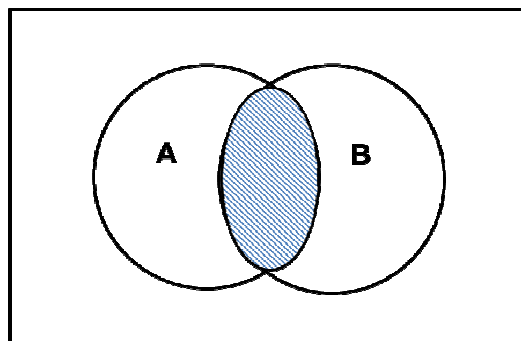


Fuente: Ríos Nicolás, 2024.

- **INNER JOIN:** Filtra dos conjuntos de resultados según un criterio común.

Figura 14.

Expresión de Conjunto INNER JOIN.

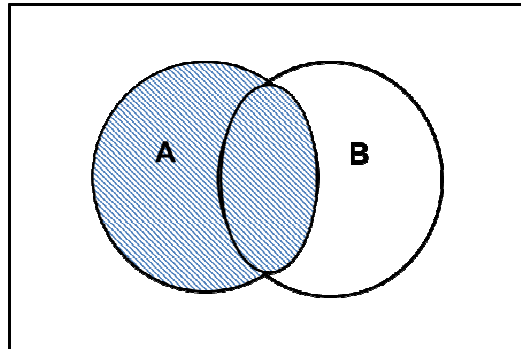


Fuente: Ríos Nicolás, 2024.

- LEFT JOIN: Vincula el conjunto de resultados de la consulta de la izquierda con los resultados coincidentes de la consulta de la derecha; los campos no coincidentes se fijan como NULL.

Figura 13.

Expresión de Conjunto INNER JOIN.

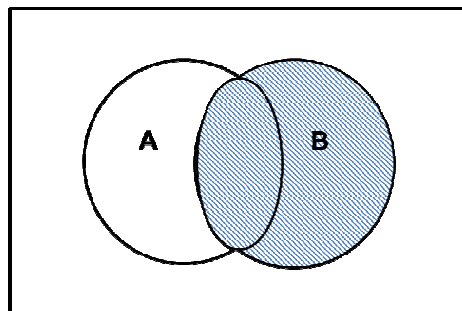


Fuente: Ríos Nicolás, 2024.

- RIGHT JOIN: Vincula el conjunto de resultados de la consulta de la derecha con los resultados que coinciden de la consulta de la izquierda; los campos que no coinciden se fijan como NULL.

Figura 13.

Expresión de Conjunto INNER JOIN



Fuente: Ríos Nicolás, 2024.

Consultas básicas en lenguaje estructurado

Probablemente la instrucción *SELECT* sea la más versátil y la más empleada. Es a la que recurrimos para obtener una información concreta de una base de datos y cuenta con la ventaja de poder consultar una sola tabla o varias.

Una vez que identificamos la función *SELECT*, debemos conocer también las cláusulas, instrucciones que se emplean para concretar cuáles son los datos que nos interesan, las condiciones de modificación que definirán los datos que vamos a seleccionar:

FROM: indica la tabla con la que se va a trabajar.

WHERE: fija las condiciones que deben cumplir los registros que se van a seleccionar.

GROUP BY: clasifica los registros seleccionados en diferentes grupos específicos.

HAVING: define la condición que debe cumplir cada grupo.

ORDER BY: ordena los registros seleccionados atendiendo a unos criterios.

La sintaxis de una consulta básica:

SELECT

Campo1 (nombre), Campo2 (apellido), Campos 3 (mail)...

FROMTabla (Clientes)

Otra consulta aplicando una cláusula:

```
SELECT
    Campo1 (nombre), Campo2 (apellido), Campos 3      (mail)...
FROM      Tabla (Clientes)
ORDER BY  (apellido)
```

Esta consulta devolverá una tabla en memoria con el nombre, apellido y mail de la tabla Clientes. Adicional éstos datos aparecerán siguiendo un orden determinado, es decir, se organizará por (apellido).

Es importante resaltar que en las consultas SQL se aplican los mismos principios del manejo de datos con los Operadores Aritméticos Lógicos y Relacional.

Operadores Aritméticos

Cuadro 1.
Operadores Aritméticos

| Operador | Descripción |
|----------|----------------|
| + | Suma |
| - | Resta |
| * | Multiplicación |
| / | División |

Fuente: Nicolás Ríos, 2024.

Operadores Lógicos

Cuadro 2.
Operadores Lógicos

| Operador | Descripción |
|----------|--|
| AND | Analiza dos condiciones y devuelve únicamente valores en los que se cumplen ambas. |
| OR | Analiza dos condiciones y devuelve valores en los que una de las dos es cierta |
| NOT | Devuelve el valor contrario de la expresión que se plantea. |

Fuente: Nicolás Ríos, 2024

Operadores Relacionales

Cuadro 3.
Operadores Relacionales

| Operador | Descripción |
|----------|-------------|
| < | Menor que |
| > | Mayor que |

Continuación Cuadro 3.
Operadores Relacionales

| | |
|---------|---|
| <> | Distinto de |
| <= | Menor ó Igual que |
| >= | Mayor ó Igual que |
| = | Igual que |
| BETWEEN | indica un intervalo de valores |
| LIKE | expresa una comparación |
| IN | especifica una relación de valores determinados |

Fuente: Nicolás Ríos, 2024

Funciones de agregado.

Las funciones de agregado o funciones agregadas aportan al lenguaje SQL utilidades de cálculo para aplicar con los datos de las tablas.

Cuadro 4.
Funciones

| Operador | Descripción |
|----------|---|
| AVG | Calcula el término medio de los valores de un campo concreto. |

Continuación Cuadro 4.

Funciones

| | |
|-------|---|
| COUNT | Halla el número de registros de la selección. |
| SUM | Suma los valores. |
| MAX | Devuelve el valor más alto. |
| MIN | Devuelve el valor más bajo. |

Fuente: Nicolás Ríos, 2024

;

.

RECURSOS INTERACTIVOS

A continuación se anexa el siguiente link que contiene un recurso audio visual para reforzar ésta primera parte de la unidad de estudio.

Enlace para descargar SQL Any 12. Manejador de Base de datos.

Incluye guía, video e instalador.

https://drive.google.com/file/d/1Kq1RjG38DPTVMYuZihU37Z-MDskHCq_b/view?usp=drive_link

Guía de Conectividad vía ODBC, (Ver Anexo C)

Actividades Sugeridas.

Actividad 6. Crear Base Datos .

- De la actividad 5, aplicando la mejora de normalización, proceda a crear la BD en la aplicación.

Tomar en cuenta: Creación de Tablas, Campos con sus respectivos tipos, tamaños y formatos, asignación de Claves Primarias, Claves Foráneas, Relaciones establecidas entre tablas, Carga de registros a las tablas.

Actividad 7. Práctica de Consultas SQL.

- Realizar consultas SQL a diferentes tablas DEMO de la aplicación. (Ver Anexo D)

REFERENCIAS CONSULTADAS

ADMINISTRACIÓN DE BASE DE DATOS.

DISEÑO Y DESARROLLO DE APLICACIONES. (2007). Tercera edición.

Michael V. Mannino. Revisión técnica: Carlos Villegas Quezada. México:
Editorial McGraw-Hill Interamericana.

Universitaria de Investigación y Desarrollo UDI

Grupo de Investigación en Nuevas Tecnologías Aplicadas a la Educación.
(2009)

Introducción a las Bases de Datos. Primera edición. Ricardo Vicente Jaime
Vivas. Bucaramanga. GIDSAW.

FUNDAMENTOS DE BASES DE DATOS. (2002). Cuarta edición.

Abraham Silberschatz, Henry F. Korth, S. Sudarshan. Traducido de la cuarta
edición en inglés de DatabaseSystemConcepts. Revisión técnica LUIS GRAU
FERNÁNDEZ. Universidad Nacional de Educación a Distancia. España.
McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.

BASE DE DATOS. (2011). Primera edición.

Mercedes Marqués - ISBN: 978-84-693-0146-3.

Edita: Publicacions de la Universitat Jaume I. Servei de Comunicació i
Publicacions

España. Col·leccióSapientia,

Consultas SQL: las consultas básicas que debes conocer.

MARKETING Y COMUNICACIÓN|11/06/2021.

UNIR - Universidad Internacional de La Rioja

<https://www.unir.net/marketing-comunicacion/revista/consultas-sql/>

GLOSARIO.

Aula En Abierto. Esta obra está bajo una licencia de Creative Commons Reconocimiento-CompartirIgual 4.0 España.

https://formacion.intef.es/tutorizados_2013_2019/pluginfile.php/37716/mod_ims_cp/content/1/index.html

Glosario de Base de datos y Comandos SQL.

<https://codigosql.com/glosario-de-base-de-datos-y-comandos-sql/>

ANEXO A

AULA CLIC. CURSOS DE INFORMÁTICA
GRATUITOS. https://www.aulacltic.es/sql/a_8_1_1.htm

Guía Tipos de Datos 1/6

| <u>Tipo de dato</u> | <u>Sinónimos</u> | <u>Tamaño</u> | <u>Descripción</u> |
|---------------------|--|---------------------------|--|
| BINARY | VARBINARY BINARY VARYING BIT VARYING | 1 byte por carácter | Se puede almacenar cualquier tipo de datos en un campo de este tipo. Los datos no se traducen (por ejemplo, a texto). La forma en que se introducen los datos en un campo binario indica cómo aparecerán al mostrarlos. |
| BIT | BOOLEAN LOGICAL LOGICAL1 YESNO | 1 byte | Valores Sí y No, y campos que contienen solamente uno de dos valores. |
| TINYINT | INTEGER1 BYTE | 1 byte | Un número entero entre 0 y 255. |
| COUNTER | AUTOINCREMENT | | Se utiliza para campos contadores cuyo valor se incrementa automáticamente al crear un nuevo registro. |
| MONEY | CURRENCY | 8 bytes | Un número entero comprendido entre - 922.337.203.685.477,5808 y 922.337.203.685.477,5807. |
| DATETIME | DATE TIME | 8 bytes | Una valor de fecha u hora entre los años 100 y 9999 |
| DECIMAL | NUMERIC DEC | 17 bytes | Un tipo de datos numérico exacto con valores comprendidos entre 1028 - 1 y - 1028 - 1. Puede definir la precisión (1 - 28) y la escala (0 - precisión definida). La precisión y la escala predeterminadas son 18 y 0, respectivamente. |

AULA CLIC. CURSOS DE INFORMÁTICA
GRATUITOS. https://www.aulaclic.es/sql/a_8_1_1.htm

Guía Tipos de Datos 2/6

| <u>Tipo de dato</u> | <u>Sinónimos</u> | <u>Tamaño</u> | <u>Descripción</u> |
|---------------------|---|--------------------|--|
| REAL | SINGLE FLOAT4 IEEE SINGLE | 4 bytes | Un valor de coma flotante de precisión simple con un intervalo comprendido entre $-3,402823E38$ y $-1,401298E-45$ para valores negativos, y desde $1,401298E-45$ a $3,402823E38$ para valores positivos, y 0. |
| FLOAT | DOUBLE FLOAT8 IEEE DOUBLE NUMBER | 8 bytes | Un valor de coma flotante de precisión doble con un intervalo comprendido entre $-1,79769313486232E308$ y $-4,94065645841247E-324$ para valores negativos, y desde $4,94065645841247E-324$ a $1,79769313486232E308$ para valores positivos, y 0. |
| SMALLINT | SHORT INTEGER2 | 2 bytes | Un entero corto entre -32.768 y 32.767 . |
| INTEGER | LONG INT INTEGER4 | 4 bytes | Un entero largo entre $-2.147.483.648$ y $2.147.483.647$. |
| IMAGE | LONG BINARY GENERAL OLEOBJECT | Lo que se requiera | Desde cero hasta un máximo de 2.14 gigabytes. Se utiliza para objetos OLE. |

AULA CLIC. CURSOS DE INFORMÁTICA
GRATUITOS. https://www.aulaclic.es/sql/a_8_1_1.htm

Guía Tipos de Datos 3/6

| <u>Tipo de dato</u> | <u>Sinónimos</u> | <u>Tamaño</u> | <u>Descripción</u> |
|---------------------|--|--|---|
| TEXT | LONGTEXT LONGCHAR MEMO NOTE NTEXT | 2 bytes por carácter. (Consulte las notas). | Desde cero hasta un máximo de 2.14 gigabytes. |
| CHAR | TEXT(n) ALPHANUMERIC CHARACTER STRING VARCHAR CHARACTER VARYING NCHAR NATIONAL CHARACTER NATIONAL CHAR NATIONAL CHARACTER VARYING NATIONAL CHAR VARYING | 2 bytes por carácter. (Consulte las notas). | Desde cero a 255 caracteres. |

AULA CLIC. CURSOS DE INFORMÁTICA
GRATUITOS. https://www.aulaclic.es/sql/a_8_1_1.htm

Guía Tipos de Datos 4/6

Notas:

- Un campo LONGTEXT se almacena siempre en el formato de representación Unicode.
- Si se utiliza el nombre del tipo de datos TEXT sin especificar la longitud opcional (TEXT(25), por ejemplo), se crea un campo LONGTEXT.
Esto permite escribir instrucciones CREATE TABLE que producirán tipos de datos coherentes con Microsoft SQL Server.
- Un campo CHAR se almacena siempre en el formato de representación Unicode, que es el equivalente del tipo de datos NATIONAL CHAR del SQL de ANSI.
- Si se utiliza el nombre del tipo de datos TEXT y se especifica la longitud opcional (TEXT(25), por ejemplo), el tipo de datos del campo es equivalente al tipo de datos CHAR.
De ese modo, se mantiene la compatibilidad con versiones anteriores para la mayoría de las aplicaciones de Microsoft Jet, a la vez que se habilita el tipo de datos TEXT (sin especificación de longitud) para la alineación con Microsoft SQL Server.
- Los caracteres de los campos definidos como TEXT (también conocidos como MEMO) o CHAR (también conocidos como TEXT(n) con una longitud específica)

➤ **AULA CLIC. CURSOS DE INFORMÁTICA**

GRATUITOS. https://www.aulaclic.es/sql/a_8_1_1.htm

Guía Tipos de Datos 5/6

se almacenan en el formato de representación Unicode. Los caracteres Unicode requieren siempre dos bytes para el almacenamiento de cada carácter.

Para las bases de datos de Microsoft Jet ya existentes que contengan principalmente datos de tipo carácter, esto puede significar que el tamaño del archivo de base de datos sea casi el doble cuando se convierta al formato Microsoft Jet 4.0.

Sin embargo, la representación Unicode de muchos juegos de caracteres, antes denominados juegos de caracteres de un solo byte (SBCS), puede comprimirse fácilmente a caracteres de un solo byte.

Si define una columna CHAR con el atributo COMPRESSION, los datos se comprimirán automáticamente a medida que se almacenen y se descomprimirán cuando se recuperen de la columna.

Los caracteres Unicode y su compresión.

En ACCESS 2000 se utiliza el formato de representación de caracteres Unicode, los caracteres Unicode requieren siempre dos bytes para cada carácter lo que permite una gama más amplia de caracteres.

Para las bases de datos de Microsoft® Jet ya existentes que contengan principalmente datos de tipo carácter, esto puede significar que el tamaño del archivo de base de datos sea casi el doble cuando se convierta al formato Microsoft Jet versión 4.0.

AULA CLIC. CURSOS DE INFORMÁTICA
GRATUITOS. https://www.aulaclic.es/sql/a_8_1_1.htm

Guía Tipos de Datos 6/6

Sin embargo, la representación Unicode de muchos juegos de caracteres, antes denominados juegos de caracteres de un solo byte (SBCS), puede comprimirse fácilmente a caracteres de un solo byte. Si se define una columna CHARACTER con el atributo WITH COMPRESSION (propiedad Compresión Unicode), los datos se comprimirán automáticamente cuando se almacenen y se descomprimirán cuando se recuperen de la columna.

Las columnas MEMO también pueden ser definidas de modo que almacenen datos en formato comprimido. No obstante, existe una restricción. Sólo se comprimirán las instancias de columnas MEMO que, tras la compresión, ocupen 4.096 bytes o menos.

El resto de instancias de columnas MEMO quedarán sin comprimir. Esto significa que, dentro de una tabla determinada, para una columna MEMO dada, algunos datos pueden estar comprimidos y otros no.

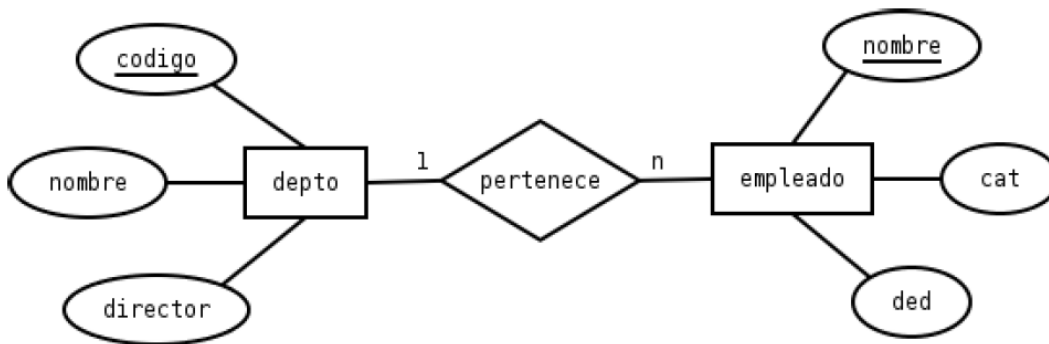
ANEXO B

EJERCICIOS DE BASES DE DATOS. SOLUCIONES

Introducción a la Informática Licenciado en ADE Fac. de Administración y Dirección de Empresas. UPV.

Ejemplo 1

Dado un código de departamento, conocer su nombre, director y los empleados de ese departamento con su nombre, categoría y dedicación.



dep

| Campo | Descripción |
|---------------|-------------|
| <u>codigo</u> | |
| nombre | |
| director | |

emplea

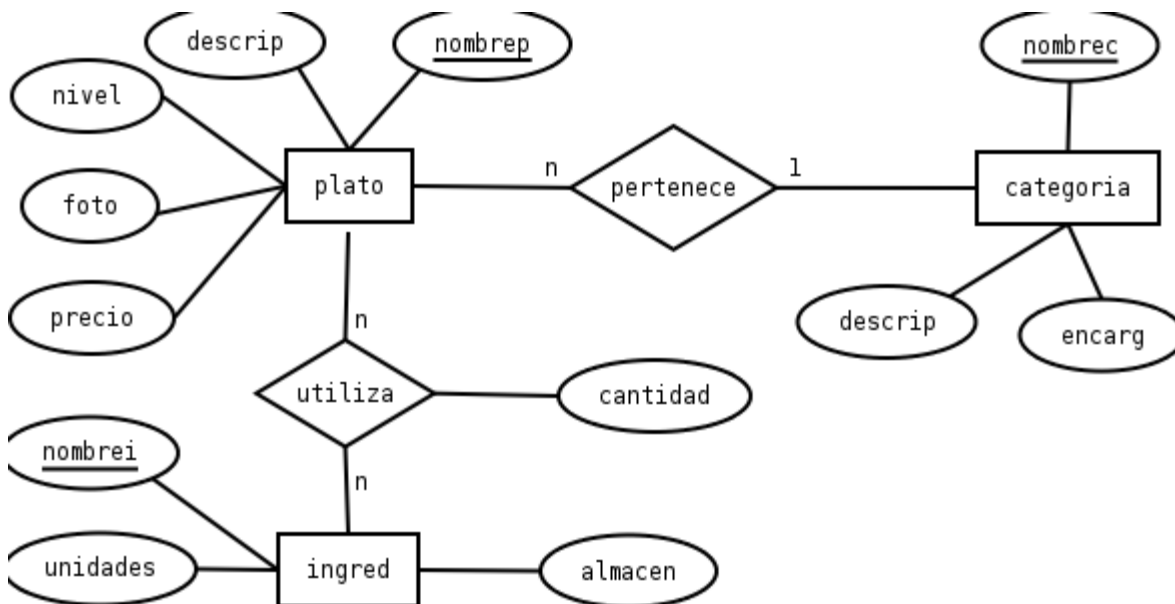
| Campo | Descripción |
|---------------|-------------|
| <u>nombre</u> | |
| cat | |
| ded | |
| codigo | |

Ejercicio 1

Se desea construir una base de datos que almacene la carta de un restaurante.

Para cada plato, se desea obtener su nombre, descripción, nivel de dificultad (de elaboración), una foto y el precio final para el cliente. Cada plato pertenece a una categoría.

Las categorías se caracterizan por su nombre, una breve descripción y el nombre del encargado. Además de los platos, se desea conocer las recetas para su realización, con la lista de ingredientes necesarios, aportando la cantidad requerida, las unidades de medida (gramos, litros, etc. . .) y cantidad actual en el almacén.

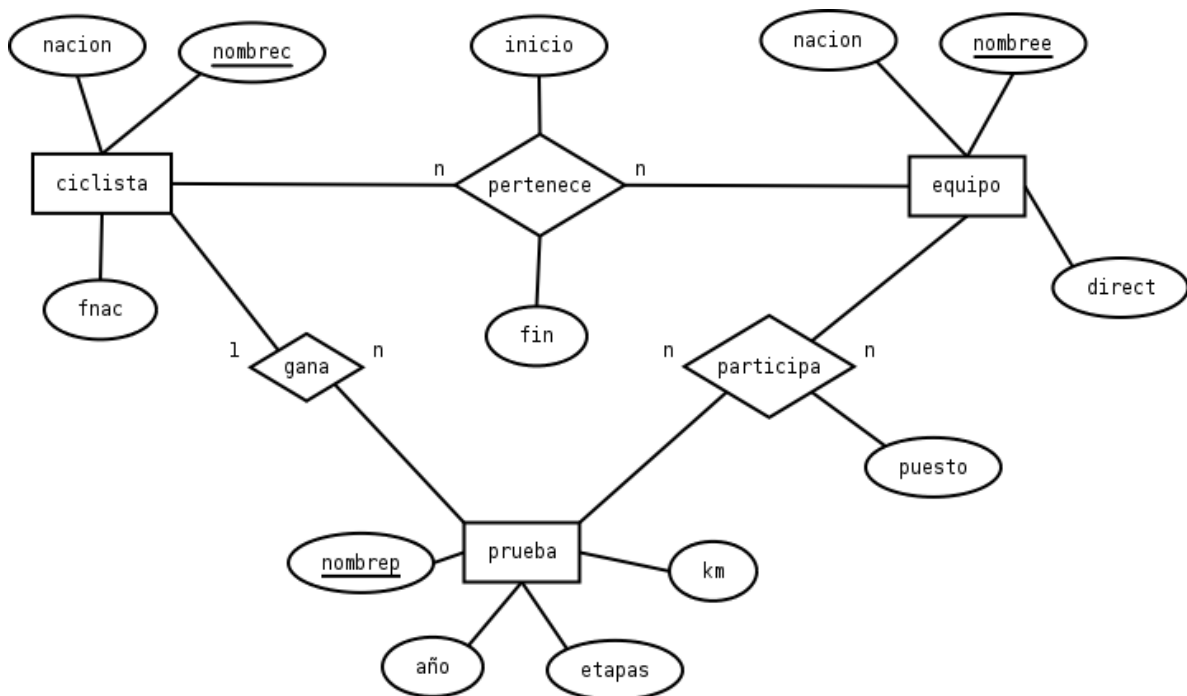


Ejercicio 2.

Con la próxima edición de la Vuelta Ciclista a España, un periódico deportivo quiere crear una base de datos para mantener información sobre las pruebas ciclistas por etapas. En la base de datos debe aparecer información sobre los ciclistas, los equipos a los que pertenecen y las pruebas en las que cada equipo ha participado (se asume que participa todo el equipo).

De cada ciclista, se desea conocer su nombre, nacionalidad y fecha de nacimiento, así como el equipo al que pertenece, manteniendo la fecha de inicio y fin de contrato con el equipo.

De cada equipo también se desea conocer su nombre, su nacionalidad, el nombre del director y las pruebas en las que ha participado, con su nombre, año de edición, nº de etapas, kilómetros totales y puesto que ocupó el equipo en la clasificación final. Un dato adicional para las pruebas es saber el nombre del ciclista que quedó ganador.

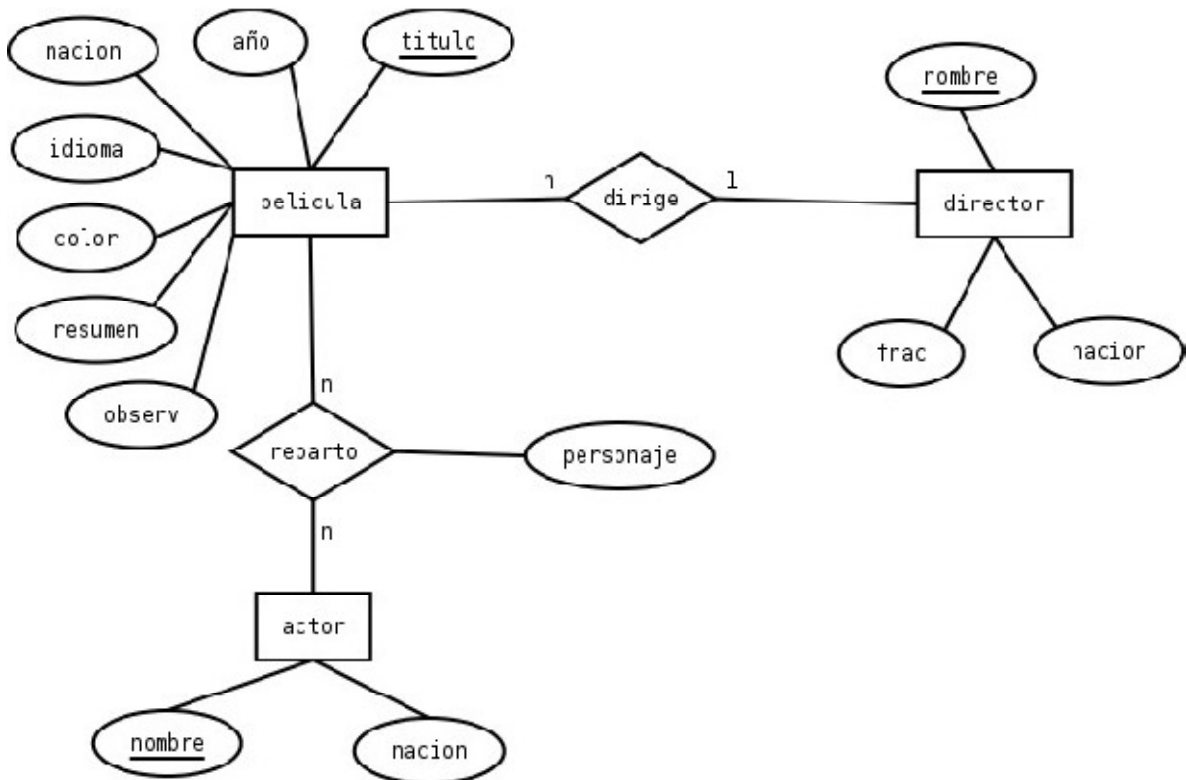


Ejercicio 3

Acabas de empezar tu colección de películas y quieres hacer una base de datos para construir su ficha técnica. De cada película, necesitas su título, año, nacionalidad y algunos datos de su director: el nombre, la fecha de nacimiento y su país de origen. Además, quieres saber su idioma, si es en blanco y negro o en color, si tiene alguna restricción de edad, un resumen y poder poner tus propias observaciones.

La ficha técnica de cada película también debe incluir el reparto de actores, donde aparecerá su nombre, su nacionalidad y el nombre del personaje que tiene en la película.

Diseñar una base de datos que se ajuste al requerimiento arriba expuesto, identificando tablas, atributos, claves principales y relaciones existentes.

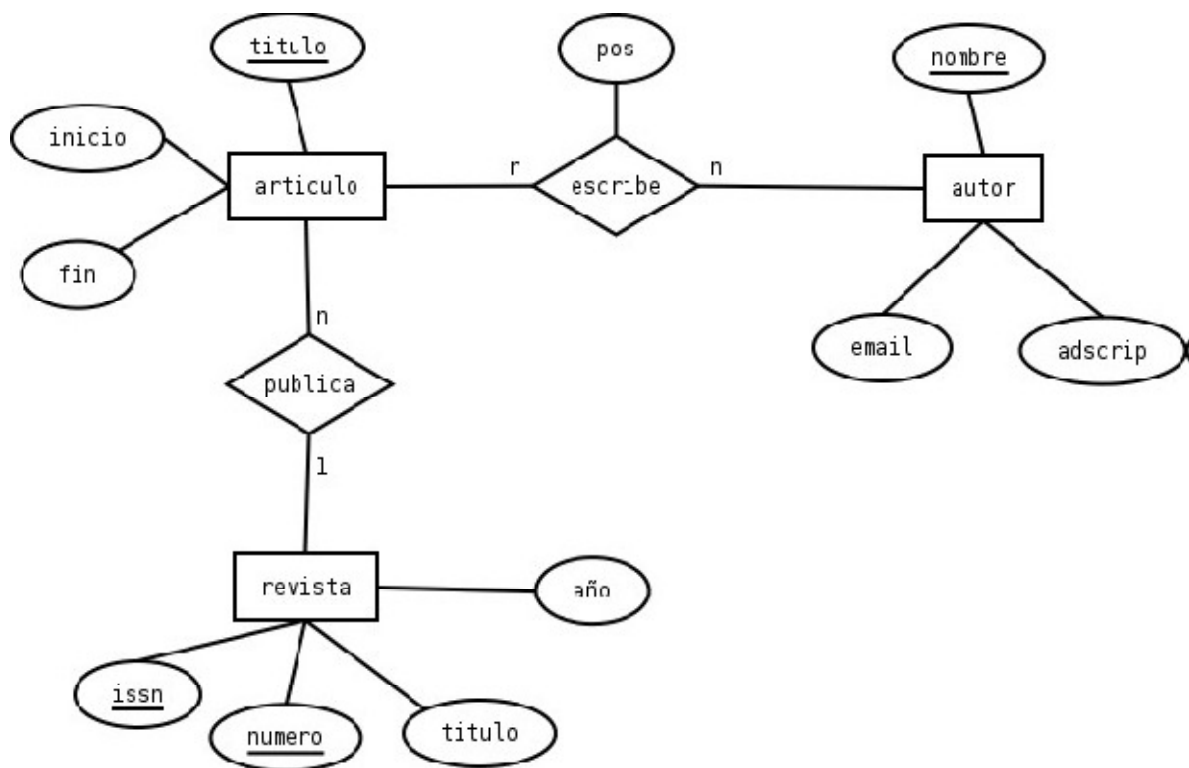


Ejercicio 4

Se desea crear una base de datos que contenga información sobre las revistas a las que estás suscrito o compras habitualmente. De cada revista, se pide su título, el ISSN (un código que identifica a la publicación), el número y el año de publicación. También se desea almacenar información de cada uno de los artículos publicados: el título, la página de inicio y la página de fin. Se asume que no hay dos artículos con el mismo título.

Cada artículo puede estar escrito por varios autores, de quienes interesa conocer su nombre, una dirección de correo electrónico y su adscripción, así como un número que indique la posición en la que aparece en cada artículo: un 1 si es el primer autor, un 2 si aparece en segundo lugar, etc.

Diseñar una base de datos que se ajuste al requerimiento arriba expuesto, identificando tablas, atributos, claves principales y relaciones existentes.

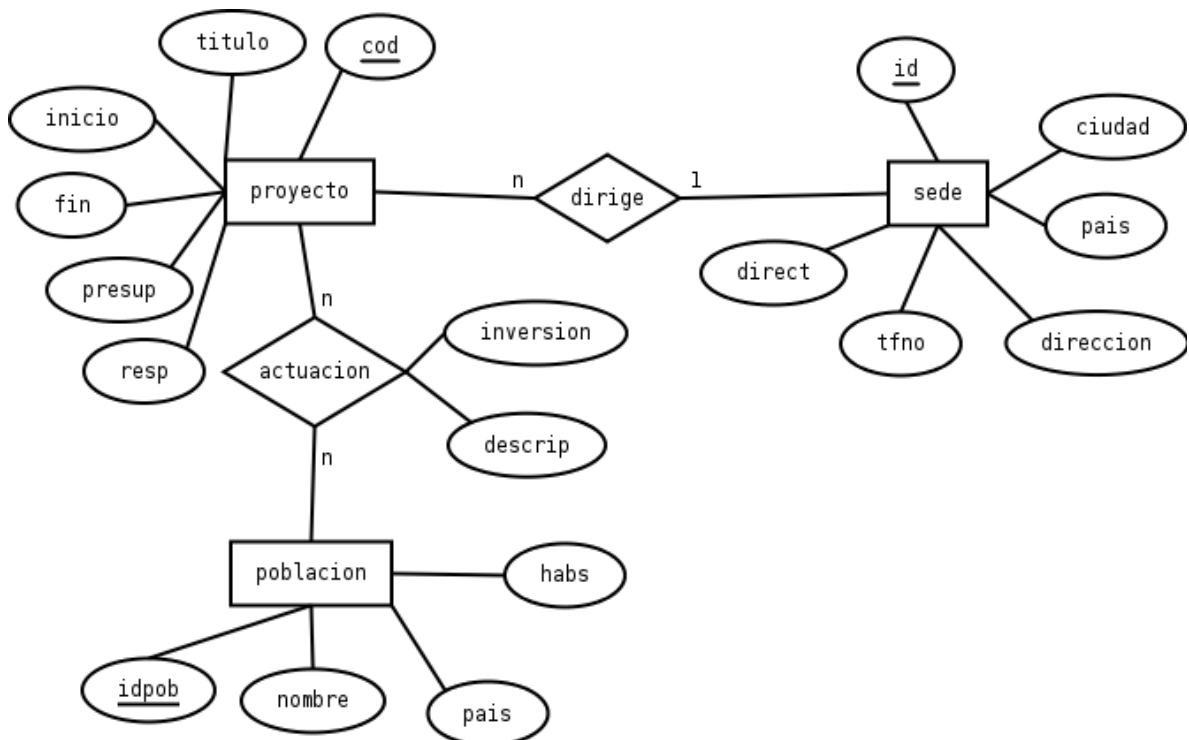


Ejercicio 5

Una ONG desea elaborar una base de datos para llevar el seguimiento de todos sus proyectos. Tiene diversas sedes en varios países que se encargan de gestionar y coordinar los proyectos de ese país, cada uno de los cuales puede afectar a una o varias poblaciones.

Sobre la sedes se desea mantener un identificador, la ciudad y país en el que se encuentra, junto con su dirección, un teléfono de contacto y el nombre del director. Cada sede gestiona un conjunto de proyectos, con un código, un título, fechas de inicio y finalización, el presupuesto asignado y el nombre del responsable.

De cada proyecto es necesario conocer qué actuaciones se realizan en cada población, almacenando el nombre, país y nº de habitantes y un identificador para diferenciarlas. Además se desea la inversión del proyecto que corresponde a la población y una pequeña descripción de la actuación.

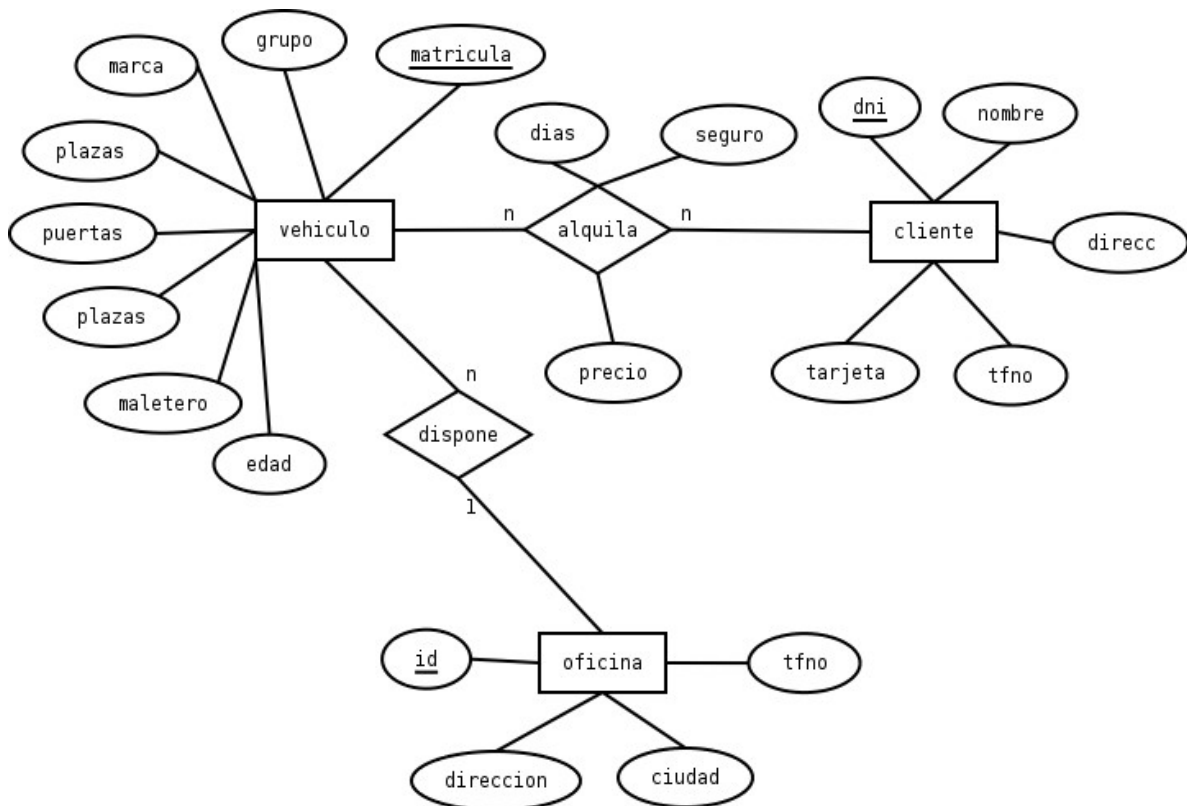


Ejercicio 6

Una empresa de alquiler de vehículos desea conocer en todo momento el estado de su flota. La empresa tiene diversas oficinas repartidas por todo el territorio español. Cada oficina se identifica por un código único y se caracteriza por la ciudad en la que se encuentra y su dirección completa (calle, número y código postal) y teléfono.

En cada oficina hay disponible un conjunto de coches, de los cuales se conoce su matrícula, el grupo al que pertenece: A, B, C, D, E, F o G (depende del tipo y tamaño del vehículo), la marca, el modelo, el número de puertas, el número de plazas, la capacidad del maletero y la edad mínima exigida para el alquiler.

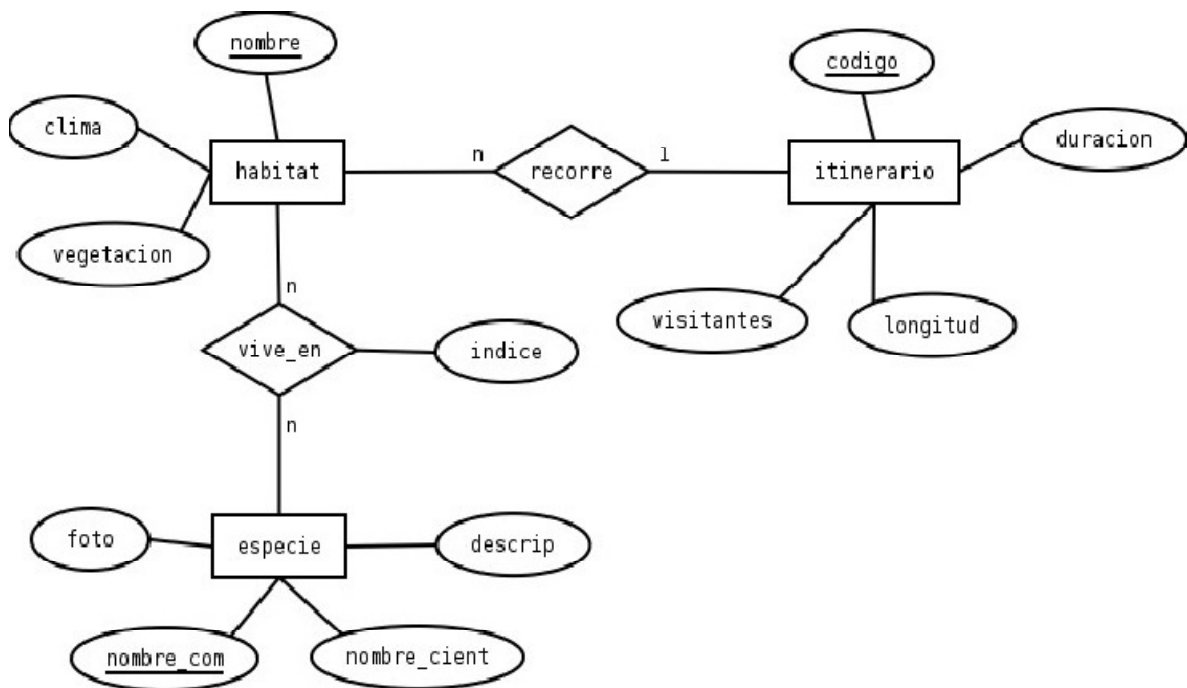
Para llevar el control del estado de cada vehículo, la empresa mantiene un registro de todos los alquileres que ha sufrido, indicando para cada uno de ellos el nombre del conductor, su DNI, su dirección, un teléfono de contacto y un número de tarjeta de crédito sobre la que realizar los cargos correspondientes. Además de esta información de los clientes, para cada alquiler se almacena su duración (en días), el tipo de seguro contratado y el precio total.



Ejercicio 7

Un parque zoológico quiere construir una BD para organizar las especies que posee y los distintos itinerarios para visitar el parque. La información se estructura de la siguiente forma. De las especies, se desea conocer su nombre común y su nombre científico, así como una descripción general y una fotografía. Cada especie puede vivir en distintos hábitats naturales, definidos por su nombre, clima y vegetación predominante. Cada especie tiene asociado un índice de vulnerabilidad dentro de cada hábitat, que mide el riesgo de extinción de la especie en el dicho hábitat.

Para organizar las visitas, y en función de los hábitats que desee recorrer un visitante, el parque le ofrece una serie de recorridos por los hábitats, que se identifican por su código y se caracterizan por su duración estimada, longitud y número máximo de visitantes permitidos. Un hábitat sólo puede formar parte de un itinerario.

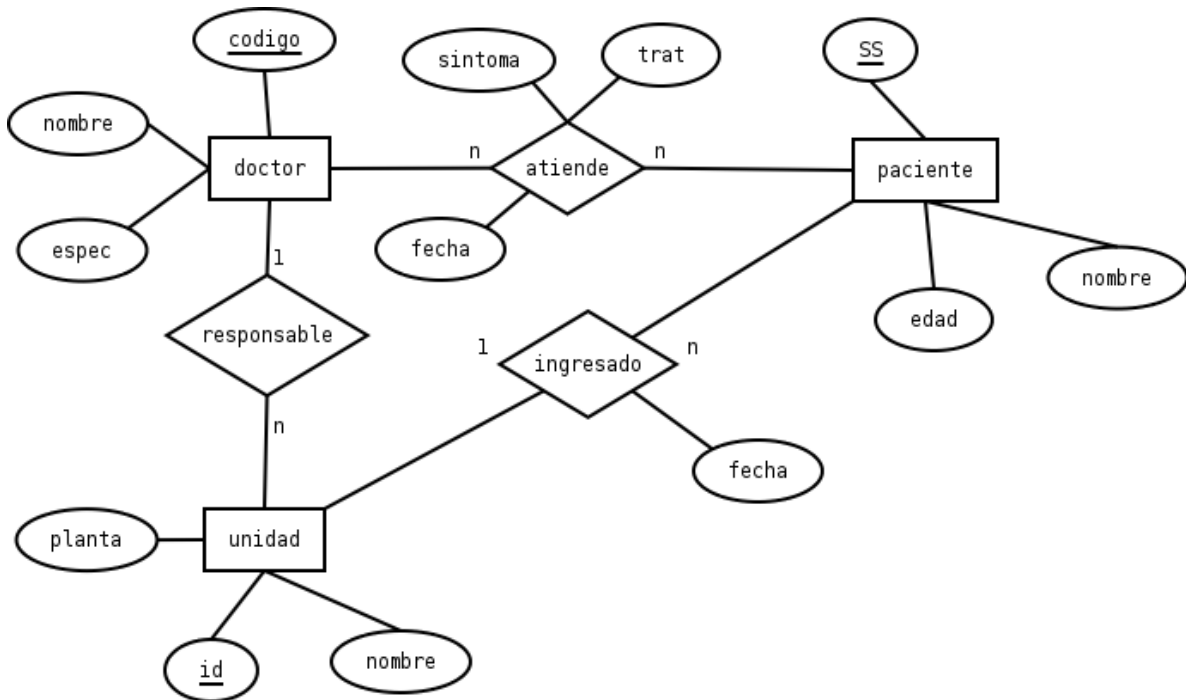


Ejercicio 8

Una clínica desea mantener una base de datos con el historial de todos los pacientes que tiene ingresados.

La clínica está dividida en varias unidades, cada una de las cuales tiene un identificador, su nombre y la planta en la que se encuentra. La unidad tiene un único doctor responsable, del cual se desea almacenar su código, el nombre y su especialidad.

Cuando llega un paciente, se le ingresa en una unidad y se registra su número de la S.S., nombre, edad y fecha de ingreso. Durante toda su estancia en la clínica, se anotan todas las intervenciones que realizan cada uno de los doctores, indicando la fecha, el síntoma observado y el tratamiento prescrito.

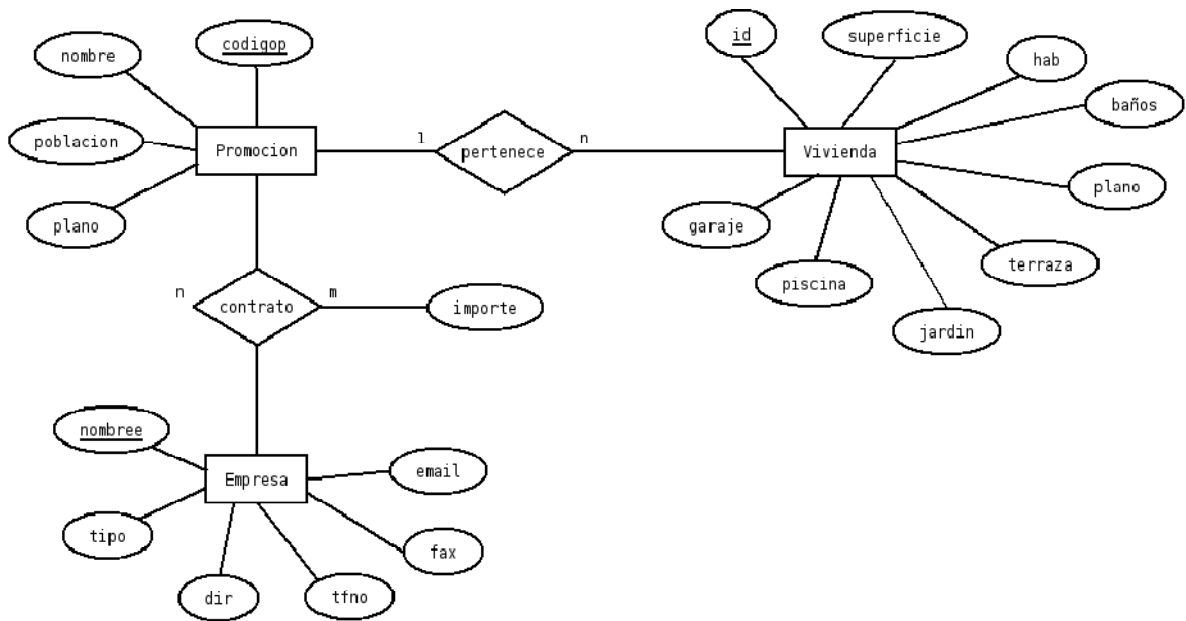


Ejercicio 9

Una promotora inmobiliaria de viviendas quiere crear una base de datos para llevar un registro de las promociones que tiene en venta. Una promoción está caracterizada por un código interno, su nombre, la población en la que está ubicada y un plano de situación.

Cada promoción está formada por un conjunto de viviendas, cada una de las cuales tiene un identificador, superficie, número de habitaciones, número de baños, el plano de la vivienda, una foto y el precio. Además es necesario indicar si tiene o no terraza, jardín privado, piscina y garaje.

Para la construcción, publicidad y venta de una promoción puede contratar a distintas empresas. De cada empresa se desea almacenar su nombre, tipo, dirección completa, teléfono, fax y dirección de correo electrónico, así como el importe del contrato entre la empresa y la promotora.

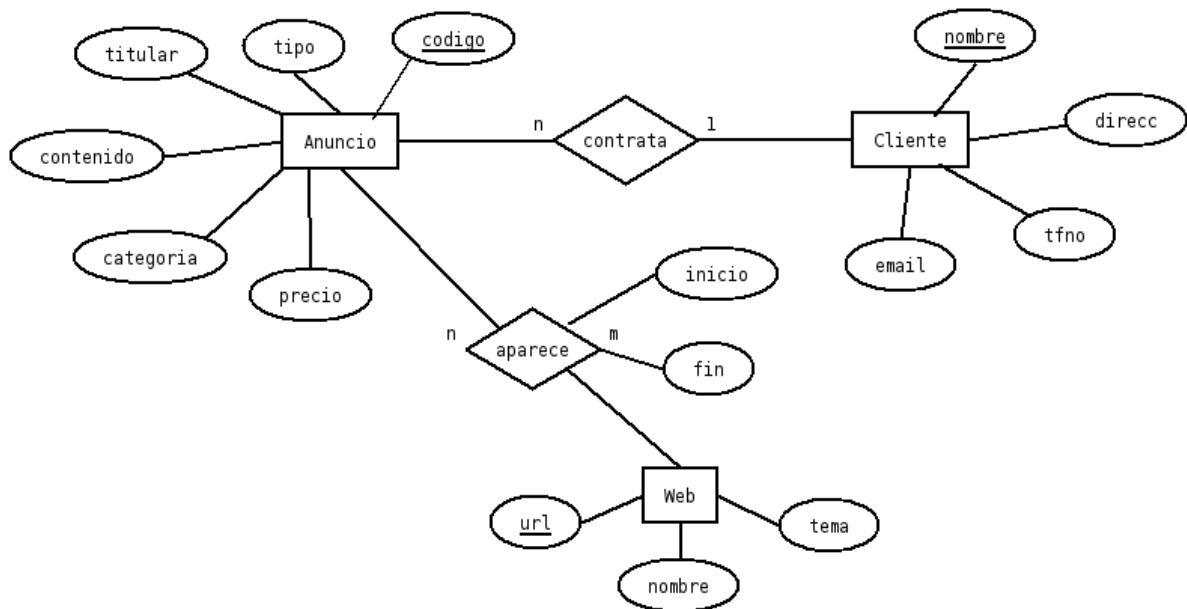


Ejercicio 10

Una agencia de publicidad necesita una base de datos para registrar todas sus campañas en la web.

Sus clientes tienen un nombre, una dirección postal, el número de teléfono y una dirección de email. Cada cliente puede contratar varios anuncios. Los anuncios quedan identificados por un código y se caracterizan por un nombre, tipo (banner, popup, enlace patrocinado,...), título, contenido, categoría y precio. Los anuncios pueden aparecer en más de una página web.

Cada web se caracteriza por su URL, nombre y tópico de interés. También se debe almacenar la fecha de inicio y de fin de la aparición del anuncio en la página web.



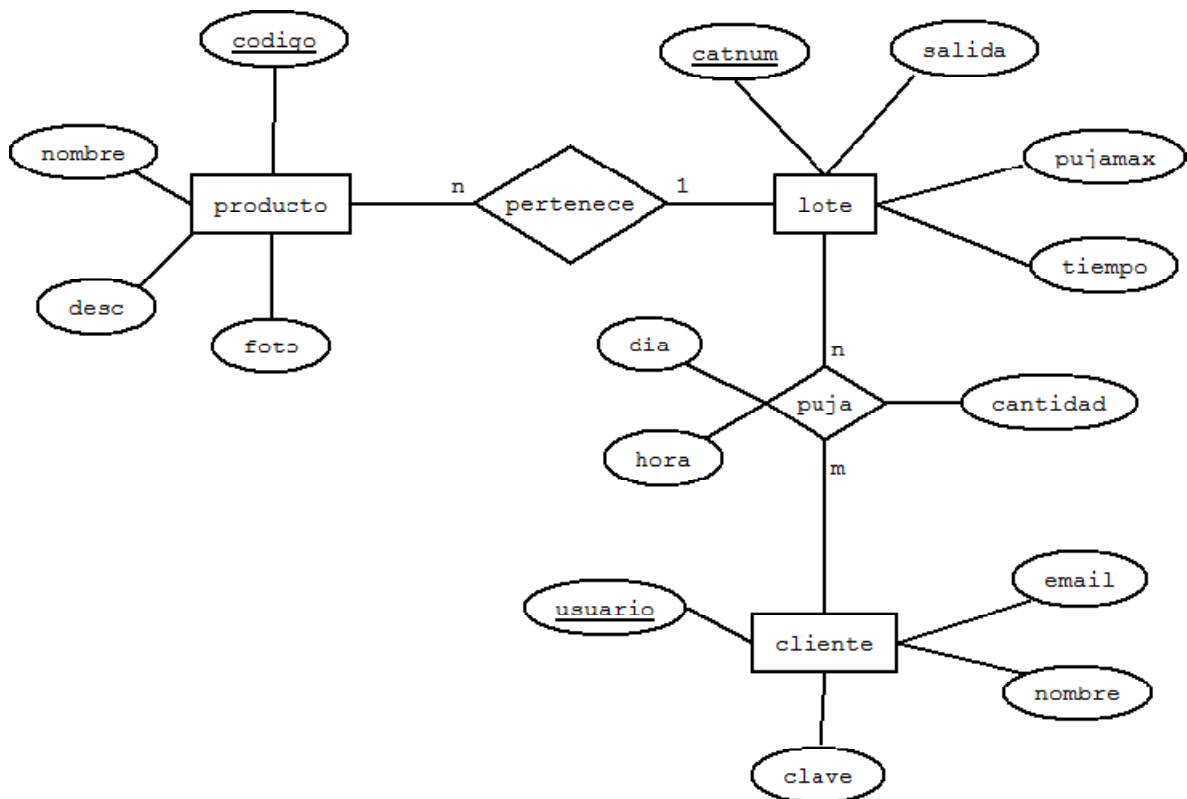
Ejercicio 11

Una casa de subastas en Internet quiere mantener una base de datos para registrar todas las transacciones que realiza.

Los productos que se subastan se agrupan en lotes. Cada lote tiene un número de catálogo, un precio de salida, la mayor puja realizada hasta el momento y el tiempo que queda de subasta. De cada producto se almacena un código, su nombre, una descripción corta y una fotografía.

Los clientes que participan en la subasta deben pujar por un lote completo (no se admiten pujas por productos individuales). Cada vez que un cliente puja, queda registrada la cantidad, el día y la hora en la que se ha producido. Para identificar a los clientes, todos deben tener un nombre de usuario, además de una contraseña, su nombre y una dirección de correo electrónico.

Construye en esquema conceptual y el esquema lógico de la base de datos que contenga la información arriba expuesta.



Ejercicio 12

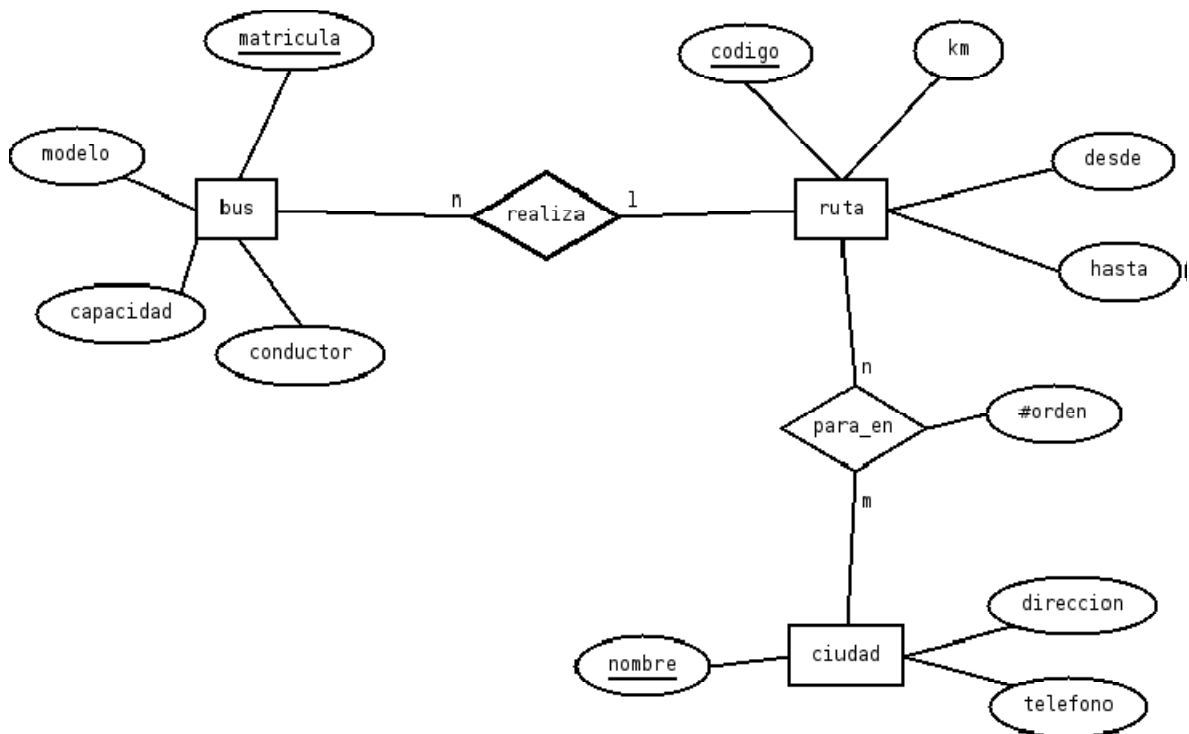
Una empresa de transporte desea crear una base de datos para almacenar información sobre sus rutas.

La empresa dispone de una flota de autobuses que distribuye en una serie de rutas. En cada ruta, el autobús pasa por un conjunto de ciudades en las que tiene parada.

Una ruta se identifica por un código y se caracteriza por los km. totales de recorrido, el origen y el destino final. De cada autobús, se almacena su matrícula, el modelo, su capacidad (plazas) y el nombre del conductor. Se asume que un autobús sólo puede estar realizando una ruta.

Las rutas tienen paradas en distintas ciudades. De cada ciudad, almacenaremos el nombre, junto con la dirección y el teléfono del lugar de parada. Para organizar las rutas, cada parada tiene un número de orden, que puede variar entre distintas rutas (una misma ciudad puede pertenecer a varias rutas).

Construye en esquema conceptual y el esquema lógico de la base de datos que contenga la información arriba expuesta.



Ejercicio 13

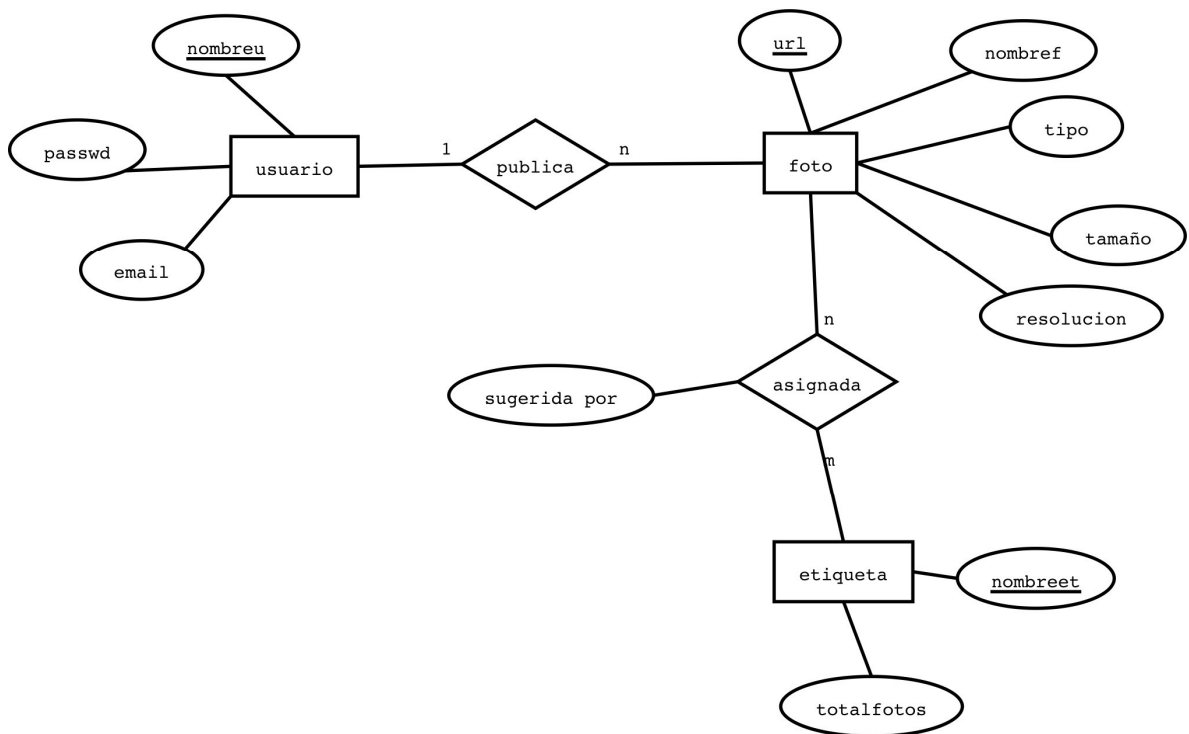
Una empresa que proporciona almacenamiento de fotos en la web necesita una base de datos para almacenar todos los archivos.

Las fotos se caracterizan por su URL (es única); además se almacena el nombre del archivo, tipo, tamaño, resolución y un campo que indica si la fotografía es pública o privada.

Los usuarios pueden añadir todas las etiquetas que necesiten a cada una de sus fotos para clasificarlas. Además del nombre de la etiqueta, se almacena el número total de fotos que la emplean. El sistema también puede sugerir etiquetas adicionales, por lo que se debe indicar quién ha sugerido la etiqueta: el usuario o el sistema. Y eso para cada foto.

Para identificar a los usuarios, se almacena el nombre de usuario, su password y su dirección de correo electrónico.

Construye el esquema conceptual y el esquema lógico de la base de datos que cumpla los requerimientos expuestos.



Ejercicio 14

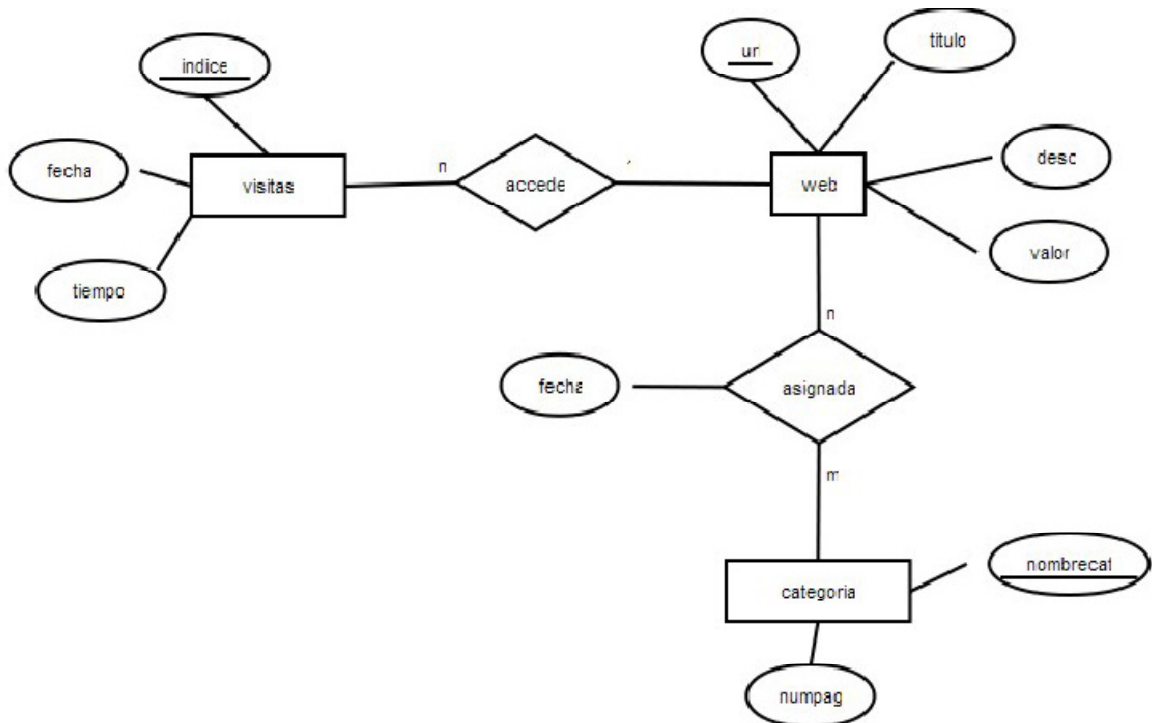
Necesitas tener una base de datos con los enlaces de las páginas web que visitas para mantener un histórico más completo que el servicio que proporcionan los Favoritos de tu navegador.

De cada página que visitas, te interesa su URL (su dirección), su nombre y una descripción breve y una valoración global.

Las páginas se van a agrupar por categorías en lugar de por carpetas, de manera que a cada página se le pueden asignar varias categorías. De cada categoría, además de su nombre, queremos saber cuántas páginas web pertenecen a ella y en qué fecha se asignó esa categoría a la página.

Para conocer el histórico de navegación, cada vez que visitemos una página web, se añadirá automáticamente a la base de datos una entrada con la fecha de consulta y el tiempo de permanencia en el sitio. Las visitas tienen asociadas un número de orden único.

Construye el esquema conceptual y el esquema lógico de la base de datos que cumpla los requerimientos expuestos.

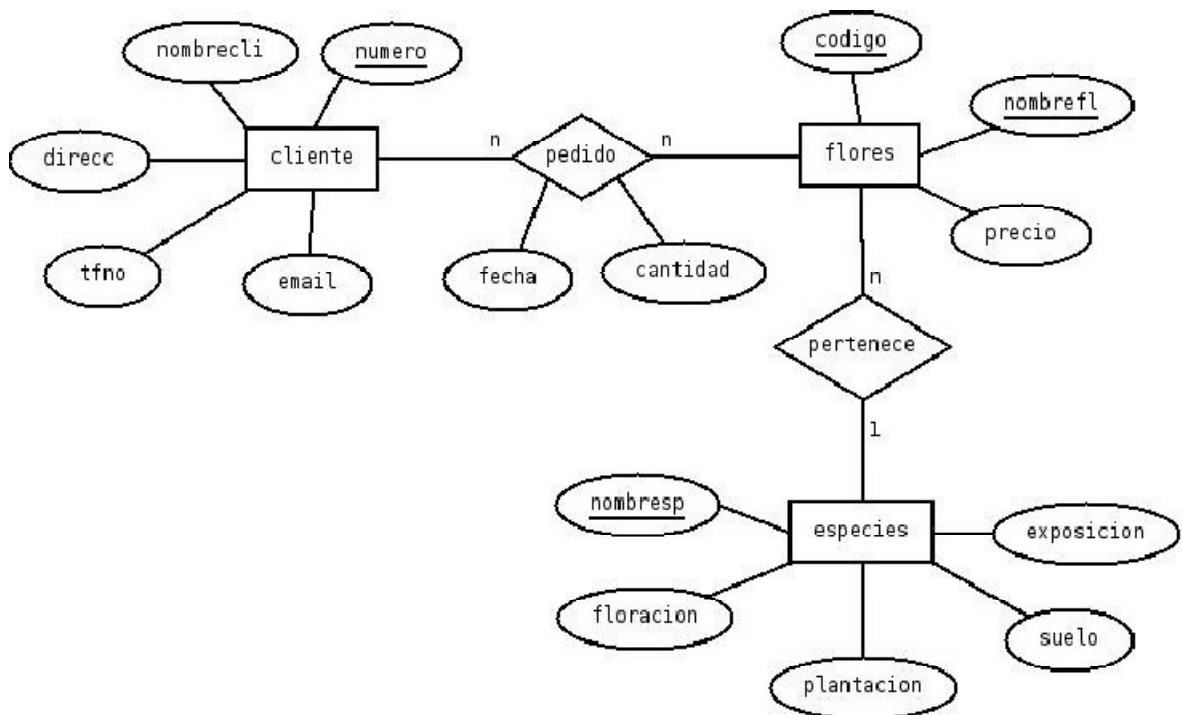


Ejercicio 15

Una pequeña floristería desea ampliar su negocio y realizar ventas a través de Internet. Y para ello necesita crear una base de datos.

Cada pedido incluye un número de pedido, la fecha de venta, el importe total y una lista con las flores solicitadas y en qué cantidad. Las flores se identifican mediante un código, su nombre y el precio de venta. Las flores pertenecen a una especie determinada. Para cada especie, se almacena el nombre, la época de floración, la estación de plantación, el tipo de suelo apropiado y el tiempo de exposición recomendado.

Construye el modelo conceptual y el modelo lógico correspondientes al enunciado expuesto



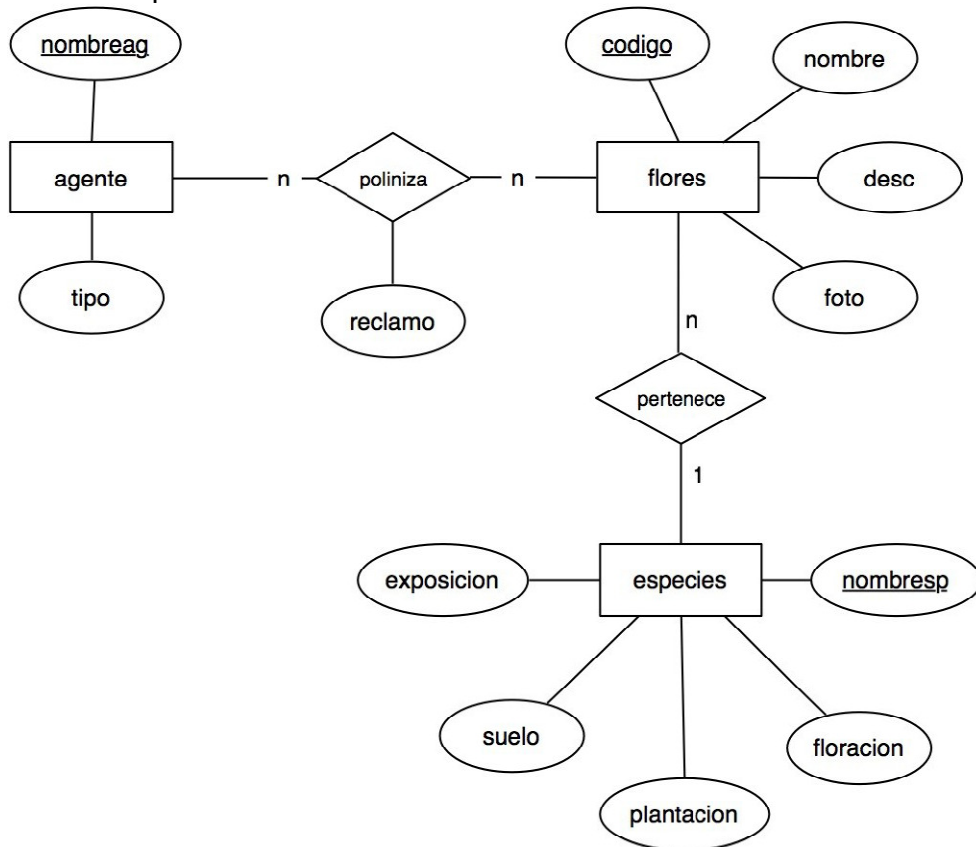
Ejercicio 16

Un centro de investigación desea recoger información sobre los procesos de polinización que se producen en una zona determinada, para lo que necesita diseñar una base de datos.

Las flores se identifican mediante un código, su nombre, una descripción y una fotografía. Las flores pertenecen a una especie determinada y para cada especie, se almacena el nombre, la época de floración, la estación de plantación, el tipo de suelo apropiado y el tiempo de exposición recomendado.

Cada flor puede ser polinizada por diversos agentes polinizadores (algunos físicos, como el viento, y otros animados, como pájaros o insectos). Para cada agente se almacena su nombre, su tipo y un subtipo. Las flores pueden usar distintos tipos de reclamo para atraer a distintos agentes (colores, olores, formas...)

Construye el modelo conceptual y el modelo lógico correspondientes al enunciado expuesto.



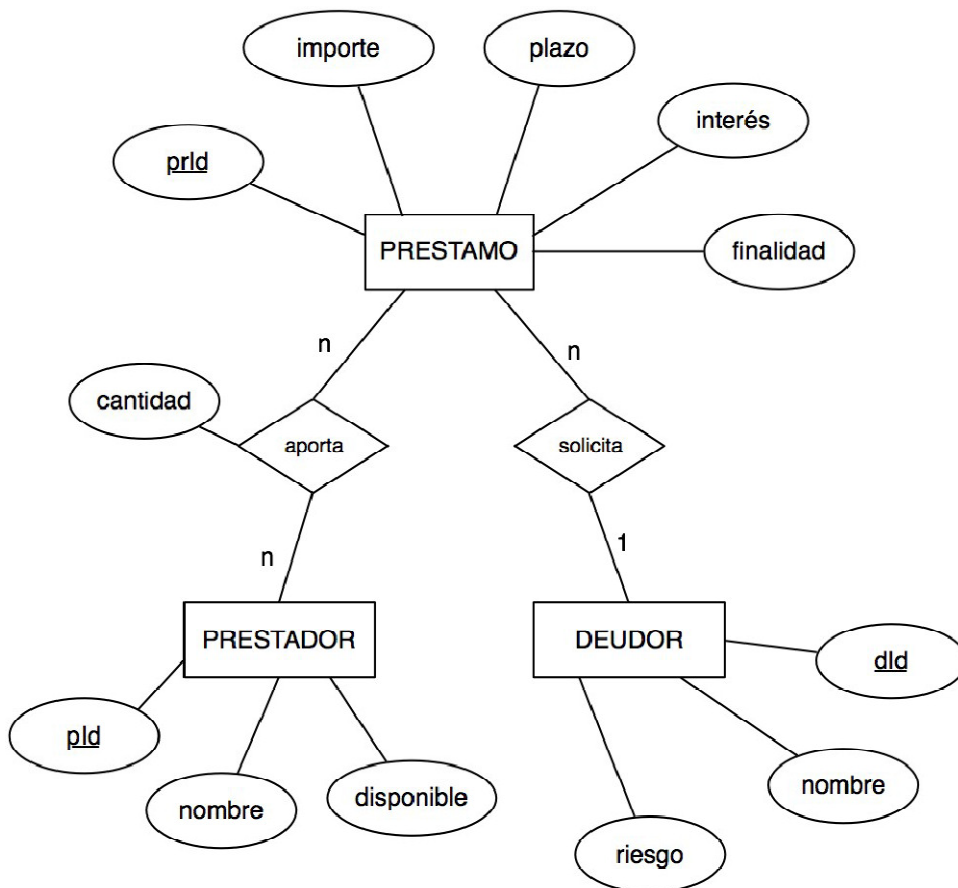
Ejercicio 17

Prestame.com es una empresa que se dedica al préstamo entre personas (p2p lending).

Los prestadores que desean prestar dinero a otros se registran con un id, su nombre y la cantidad de dinero disponible para las operaciones. Los deudores se identifican por su id y además el sistema almacena su nombre y un valor de riesgo en función de su situación personal.

Cuando el deudor solicita un préstamo, se añade un código de préstamo, el importe total, el plazo de devolución, su interés y la finalidad del mismo. Los prestadores indican qué cantidad quieren aportar a un préstamo. Un prestador puede aportar distintas cantidades parciales a varios préstamos.

Construye en el modelo conceptual y el modelo lógico correspondientes al enunciado expuesto.



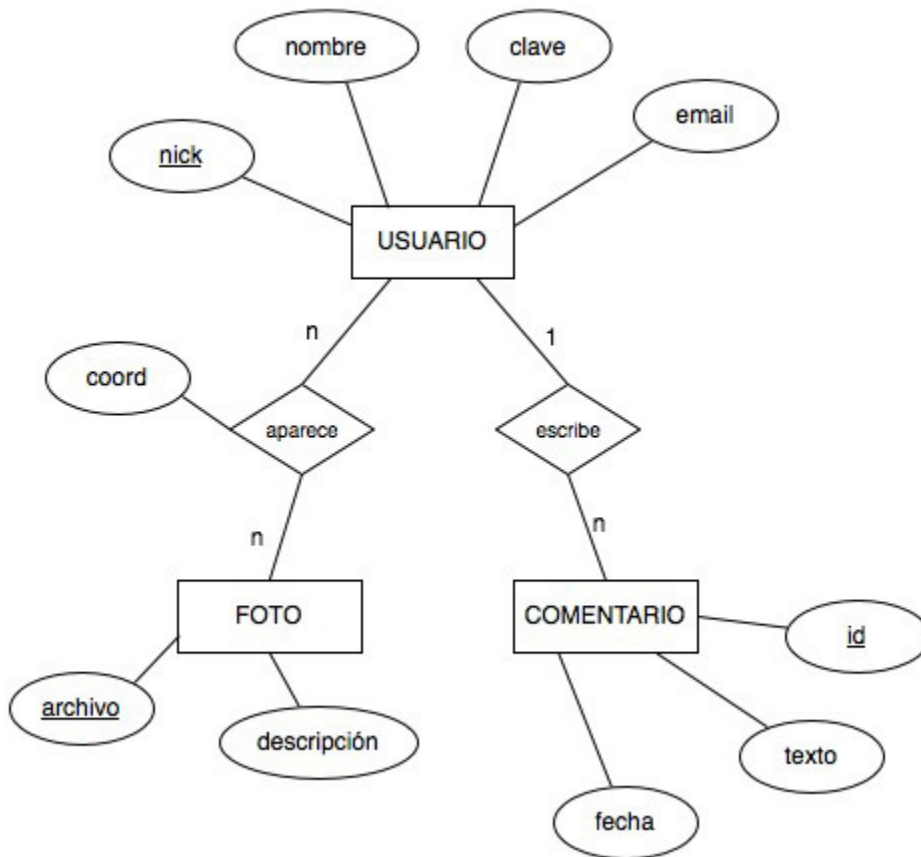
Ejercicio 18

En redes sociales como Facebook los perfiles de usuario son una herramienta fundamental. Crea una base de datos sencilla que permita almacenarlos.

Los usuarios tienen un nombre real, su nick (que es único), una clave de acceso y una cuenta de correo electrónico. Los usuarios pueden escribir comentarios, los cuales están formados por un texto y la fecha de creación. Además, los usuarios pueden aparecer en fotografías.

Una fotografía tiene un nombre de archivo y una descripción opcional. Para marcar a los usuarios en las fotografías, se emplea un marco del que se guardan sus coordenadas.

Construye en el modelo conceptual y el modelo lógico correspondientes al enunciado expuesto.



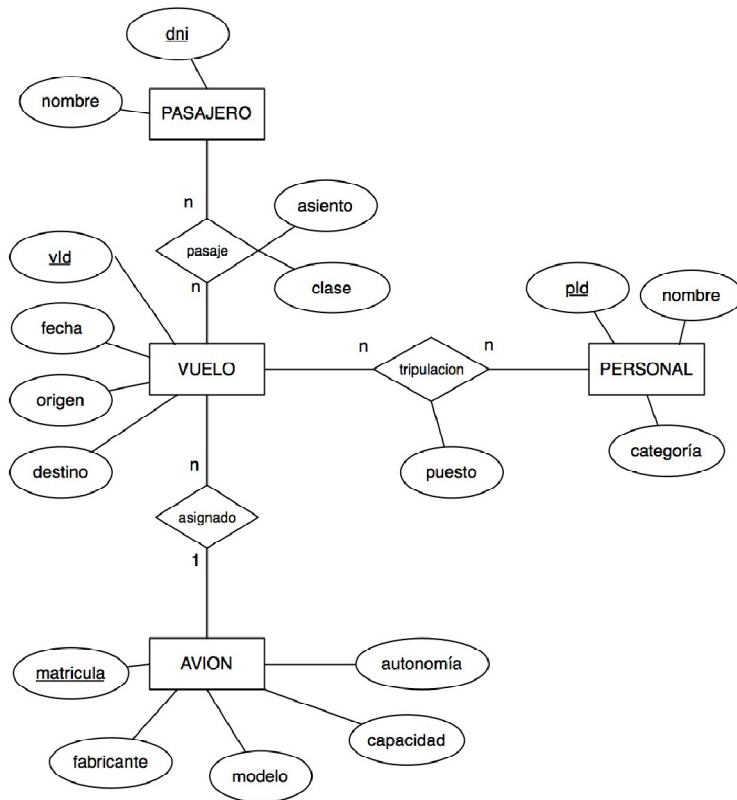
Ejercicio 19

Una compañía aérea necesita una base de datos para registrar la información de sus vuelos.

Los vuelos están caracterizados por un Id, la fecha y los aeropuertos de origen y destino. Cada vuelo es realizado por un avión. Los aviones tienen una matrícula que los identifica, el fabricante, un modelo e información sobre su capacidad (número máximo de pasajeros) y autonomía de vuelo (en horas). La tripulación asignada al vuelo está formada por el personal de la propia compañía. De cada trabajador se conoce su id, su nombre y su categoría profesional, así como el puesto que ocupa en cada vuelo en particular.

Por último, para cada vuelo, se almacena la lista completa de pasajeros, con su dni, el nombre, el asiento que ocupa y su clase (turista, primera o business).

Construye en el modelo conceptual y el modelo lógico correspondientes al enunciado expuesto.

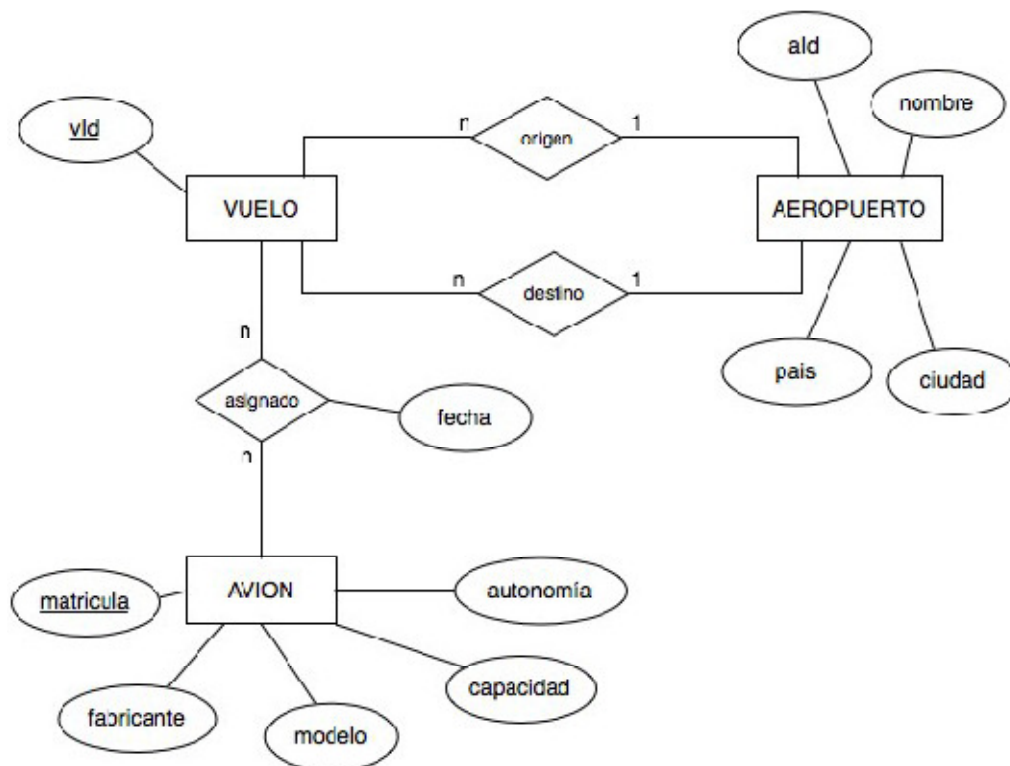


Ejercicio 20

Una compañía aérea necesita una base de datos para registrar la información de sus vuelos.

Los vuelos tienen un identificador único. Además, cada vuelo tiene asignado un aeropuerto de origen y uno de destino (se asume que no hay escalas). Los aeropuertos están identificados por unas siglas únicas (por ejemplo: VLC-Valencia, BCN-Barcelona, MAD-Madrid). Además, de cada aeropuerto se guarda el nombre de la ciudad en la que está situado y el país. Cada vuelo es realizado por un avión. Los aviones tienen una matrícula que los identifica, el fabricante, un modelo e información sobre su capacidad (número máximo de pasajeros) y autonomía de vuelo (en horas). La asignación de aviones a vuelos no es única, así que es necesario saber la fecha en la que un avión realizó cada uno de los vuelos asignados.

Construye en el modelo conceptual y el modelo lógico correspondientes al enunciado expuesto



ANEXO C

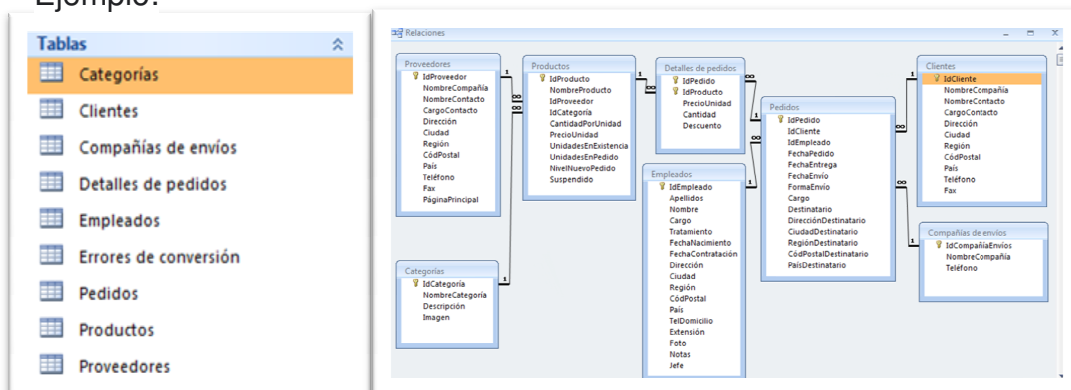
LA CONECTIVIDAD ABIERTA DE BASE DE DATOS (ODBC)

OPEN DATABASE CONNECTIVITY

ODBC es una interfaz de nivel de llamada que permite que las aplicaciones tengan acceso a los datos de cualquier base de datos en la que haya un controlador ODBC.

La utilización de ODBC permite crear aplicaciones de base de datos con acceso a cualquier base de datos en la que el usuario final tenga un controlador ODBC.

Ejemplo:



Abrir: Panel de Control → Herramientas Administrativas → Orígenes de datos ODBC

DNS del Sistema, Agregar (clic), Seleccionar Driver do Microsoft Access (*.mdb) Finalizar.

Colocar el Nombre de Referencia del Origen de Datos, agregar descripción (opcional).

Base de Datos: Seleccionar la ruta donde se encuentra la BD. Aceptar.

Conectarse a Interactive SQL.

Action: Connect with ODBC Data Source → Browse... (clic)

En la lista selecciona el creado. Aceptar. Connect.

ANEXO D

PRÁCTICA DE CONSULTAS SQL

Conectarse a la Base de Datos Demo:

ODBC Data Source name: SQL Anywhere 12 Demo

Caso de Estudio 1

Se requiere un informe de los Clientes (Customers) que se encuentren en Canadá, que muestre nombre compañía, ciudad , estado , país.

Por otra parte, mostrar las personas contactos (Contacts) nombre, apellido, título, Ciudad, estado, país.

Caso de Estudio 2

Se solicita listar el siguiente informe:

Todos los Clientes (Customers) que se encuentren en NY, NJ, FL mostrando el Nombre de la Compañía, Ciudad, Estado, Dirección. Organizado alfabéticamente por compañías. Agregando respectivamente las personas contactos (Contacts) nombre, apellido, título, Ciudad, estado, país.

Caso de Estudio 3

Es solicitado un informe de todos los empleados (employees) del departamento (departments) de Ventas , que muestre Nombre, apellido, id del dpto., nombre dpto., ciudad, estado, país. Agrupando por el MANAGER ID y ordenando luego por estado y ciudad.

Caso de Estudio 4

El directivo desea conocer las ordenes de ventas (SalesOrders) emitidas por la región Eastern mostrando la fecha, nombre de cliente (Customers) Nombre de la Compañía, Ciudad, Estado, Dirección. Ordenado por clientes.

Caso de Estudio 5

La gerencia , requiere un listado de Vendedores (employees) del equipo de Ventas y Marketing, que muestre Nombre, apellido, departamento, ciudad, estado, país, Dirección. Organizando por ubicación de estado.

Caso de Estudio 6

Un vendedor, quiere conocer todas las ventas (SalesOrders) realizadas con sus respectivos Clientes(Customers) . Mostrar información relevante, el nombre del vendedor es: Clark Alison

Caso de Estudio 7

Mostrar todas las ventas (SalesOrders) realizada en las regiones Central, South mostrando los productos vendidos (SalesOrderItems). Fecha, orden , numero de línea, Cod.producto, nombre del producto y cantidad.