**DISEÑOS DE SITIOS WEB**

**301122**

**FASE 2 DISEÑO - CREAR EL GUIÓN Y LA MAQUETACIÓN PARA UN OVI**

**ENYO JOSÉ HUETTO CERVANTES - 72054761**

**GRUPO 43**

**TUTOR**

**JOSUE IGNACIO OCHOA**

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA**

**CEAD SANTA MARTA**

**MARZO DE 2019**

**ACTIVIDAD FASE DE PLANEACION Y ANALISIS**

**CURSO DISEÑOS DE SITIOS WEB - COD. 301122**

**FORMATO GUION SITIO WEB DEL OVI**

**Administración de bases de datos**

**Diseñado Por: Enyo José Huetto Cervantes**

A continuación se presenta el formato de Guion para el desarrollo de la actividad de la Fase de Planeación y Análisis, revise muy bien las instrucciones para que realice un correcto diligenciamiento del mismo.

¡Éxitos!!!

1. **Objetivos del OVI (**describa mediante el registro de 1 objetivo general y tres específicos para que se construye este OVI**)**

**Objetivo general:**

Fundamentar los conceptos de las bases de datos relacionales para su análisis y diseño lógico, y desarrollar en el estudiante las aptitudes y las actitudes que le permitan analizar, comprender y diseñar el modelo lógico de datos en las organizaciones y aplicarlo en un SGBD.

**Objetivo específico 1:**

Fundamentar los conceptos de las bases de datos relacionales para su análisis y diseño lógico planteados mediante la técnica entidad-relación.

**Objetivo específico 2:**

Construir un modelo de Base de Datos y Comprender la estructura del lenguaje SQL.

**Objetivo específico 3:**

Realizar consultas básicas (SELECT, INSERT, UPDATE, DELETE) y modificar datos.

1. **Contenido informativo del OVI por secciones (**Replique el siguiente cuadro de acuerdo al número de secciones que vaya a crear en el OVI**)**

|  |
| --- |
| **Nombre de la sección que se creara en el OVI: INTRODUCCIÓN** |
| **2.1 Objetivo de la sección: (**Registre a continuación el objetivo que tiene esta sección**)** |
| Darle al estudiante una idea del contenido del curso, así como la importancia de este en las tareas diarias de un administrador de base de datos. |
| **2.2 Recursos de consulta que usara en la sección: (**coloque el nombre del material que usara para crear los contenidos de la sección y el enlace de descarga de los mismos sean estos Texto, Imágenes, Audios o Vídeos**)** |
| **Texto: Verdana 12** |
| **2.3 Redacte un borrador del contenido de lectura en formato de texto que tendrá la sección: (**Sea este la presentación de la sección, el contenido o ambos; redacte un borrador del texto que publicara como contenido en la sección coloque un subtítulo para identificar si corresponde a la presentación de la sección o el contenido de lectura de la sección**)** |
| La información hoy en día es el motor de toma de decisiones y el tesoro más preciado de las organizaciones, ya que al conocer de una manera adecuada y de una forma acertada los procesos internos y externos se puede en determinado momento ayudar a tomar decisiones, esto con fin de lograr una mejor producción posición en el mercado.  El manejo de gran cantidad de datos es la consecuencia del aumento de información que se maneja en el transcurso de nuestras vidas y más aún en el mundo empresarial, al incrementar todo este volumen de información que diariamente se puede acumular en el manejo de una empresa u organización se hace necesario organizarla para poder encontrar resultados rápidos y óptimos en el momento de utilizarla.  El curso Administración de bases de datos tiene como estrategia, el aprendizaje basado en proyectos y estudio de casos. El estudiante durante el desarrollo del curso debe analizar, diseñar e implementar un Caso Hipotético y como proyecto final debe realizar el anteproyecto y el desarrollo de éste con una problemática del entorno de un problema real. Con estas dos estrategias se busca que el estudiante con el desarrollo del caso de estudio, conceptualice las técnicas y herramientas para que adquiera las competencias mínimas y así enfrentarse a un problema real y desarrollar las competencias en más alto grado.  En bases de datos es muy importante la planificación y la buena estructuración que se manejan al momento de crearlas lógicamente para contar con un buen diseño en cuanto al manejo de la información de manera organizada y bien relacionada. Es por esta razón la importancia del modelado de datos precisando con la técnica Entidad - Relación que nos ayudaran a diseñar aplicaciones de bases de datos más profundas y menos redundantes haciendo mejor uso de la información.  Una vez realizado el Modelo lógico del estudio realizado con la empresa u organización, es necesario pasarlo al Modelo Relacional y así comenzar a darle forma a la base de datos y aplicar los conocimientos adquiridos sobre el Modelo Relacional. Luego la generación del archivo DDl y la creación de tablas en el componente de Oracle propuesto para los trabajos del curso. |

|  |
| --- |
| **Nombre de la sección que se creara en el OVI: Sistemas de Gestión de Base de Datos (SGBD)** |
| **2.1 Objetivo de la sección: (**Registre a continuación el objetivo que tiene esta sección**)** |
| Dar a conocer al estudiante la conceptualización e implementación de los SGBD. |
| **2.2 Recursos de consulta que usara en la sección: (**coloque el nombre del material que usara para crear los contenidos de la sección y el enlace de descarga de los mismos sean estos Texto, Imágenes, Audios o Vídeos**)** |
| **Texto: Verdana 12** |
| **2.3 Redacte un borrador del contenido de lectura en formato de texto que tendrá la sección: (**Sea este la presentación de la sección, el contenido o ambos; redacte un borrador del texto que publicara como contenido en la sección coloque un subtítulo para identificar si corresponde a la presentación de la sección o el contenido de lectura de la sección**)** |
| Los SGBD son paquetes de software muy complejos y sofisticados que deben proporcionar los servicios para el buen funcionamiento de la base de datos. No se puede generalizar sobre los elementos que componen un SGBD ya que varían mucho unos de otros. Sin embargo, es muy útil conocer sus componentes y cómo se relacionan cuando se trata de comprender lo que es un sistema de bases de datos.  1. Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un SGBD y por supuesto, el SGBD debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).  2. Un SGBD debe proporcionar un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios. Este catálogo es lo que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos). Normalmente, un diccionario de datos almacena: Nombre, tipo y tamaño de los datos.   * Nombre de las relaciones entre los datos. * Restricciones de integridad sobre los datos. * Nombre de los usuarios autorizados a acceder a la base de datos. * Esquema externo, conceptual e interno, y correspondencia entre los esquemas. * Estadísticas de utilización, tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.   3. Un SGBD debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos. Una transacción en el sistema informático de la empresa inmobiliaria sería dar de alta a un empleado o eliminar un inmueble. Una transacción un poco más complicada sería eliminar un empleado y reasignar sus inmuebles a otro empleado. En este caso hay que realizar varios cambios sobre la base de datos. Si la transacción falla durante su realización, por ejemplo, porque falla el hardware, la basede datos quedará en un estado inconsistente. Algunos de los cambios se habránhechoy otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.  4. Un SGBD debe proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es el permitir que varios usuarios tengan acceso concurrente a los datos que comparten. El acceso concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está actualizando datos, pueden interferir de modo que se produzcan inconsistencias en la base de datos. El SGBD se debe encargar de que estas interferencias no se produzcan en el acceso simultáneo.  5. Un SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe. Como se ha comentado antes, cuando el sistema falla en medio de una transacción, la base de datos se debe devolver a un estado consistente. Este fallo puede ser a causa de un fallo en algún dispositivo hardware o un error del software, que hagan que el SGBD aborte, o puede ser a causa de que el usuario detecte un error durante la transacción y la aborte antes de que finalice. En todos estos casos, el SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos llevándola a un estado consistente.  6. Un SGBD debe proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.  7. Un SGBD debe ser capaz de integrarse con algún software de comunicación. Muchos usuarios acceden a la base de datos desde terminales. En ocasiones estos terminales se encuentran conectados directamente a la máquina sobre la que funciona el SGBD. En otras ocasiones los terminales están en lugares remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red. En cualquiera de los dos casos, el SGBD recibe peticiones en forma de mensajes y responde de modo similar. Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos. Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él para que el sistema sea comercialmente viable.  8. Un SGBD debe proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realizan sobre estos datos, sigan ciertas reglas. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados. Se puede considerar como otro modo de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones. La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar. Por ejemplo, se puede establecer la restricción de que cada empleado no puede tener asignados más de diez inmuebles. En este caso sería deseable que el SGBD controlara que no se sobrepase este límite cada vez que se asigne un inmueble a un empleado.  **“Planificación de proyectos de análisis, diseño e implementación de bases de datos”**  Para la planificación de un proyecto de análisis, diseño e implementación de bases de datos, es necesario tener claro el área de la organización que se desea estudiar, pues éstas deben tener un contexto delimitado. Una vez definido lo anterior, se debe planifica todo lo concerniente al levantamiento de información, pues en ésta, la participación del usuario es de vital importancia; lo mismo que la documentación de políticas y normas que tiene la organización, en el área bajo estudio. Con respecto al análisis, este se realizará con la técnica Entidad Relación, con base al levantamiento de información. El diagrama resultante, debe estar avalado por el área usuaria, que en última instancia es el modelo lógico de datos. En cuanto al diseño físico, es el resultado de pasar el modelo lógico de datos a tablas, con sus respectivos llaves principales y llaves foráneas. Por último, la implementación del diseño físico se realizará en la herramienta MySql (SGBD).  a) Levantamiento de información Para desarrollar esta fase del proyecto, se recomienda que se recolecte la información a través de entrevistas y recolección de formatos que se manejen en el área.  Para el caso de las entrevistas, se recomienda seguir los siguientes pasos:   * Seleccionar los diferentes procesos que tiene el área bajo estudio. * Seleccionar las personas que mejor conocen los anteriores procesos. * Realizar con las personas anteriores, el cronograma para la realización de la entrevista. * Realizar la entrevista, ésta no debe durar más de dos horas, si esto ocurre, se debe realizar en otra sesión. * Editar la entrevista. Por lo general, la edición toma el doble de tiempo que duró la entrevista. Aquí, se debe quitar todos los saludos, palabras o frases que no tenga que ver con el tema. * Validación de la entrevista: Aquí, se envía al usuario el texto de la entrevista, con el fin de que verifique que lo que está escrito es lo correcto. Por lo general, los usuarios quitan y colocan cosas nuevas, esto debido a que a veces desean incluir mejoras; de ahí, que la validación es otra entrevista, pero con base en el documento editado. * Los pasos 5 y 6 se repiten, tantas veces como sea necesario, hasta que el usuario le dé el visto bueno definitivo al documento. * Para el caso de los formatos, se recomienda describir cada uno de ellos y colocar en que procesos interviene.   b) Análisis de la información Para proyecto de bases de datos, el análisis consiste en realizar el modelo de datos, con base en el levantamiento de información. Es así, como en el texto se deben identificarse los conjuntos de datos, las relaciones y los atributos. Una vez identificado y realizado el diagrama, este debe validarse con los usuarios del área.  c) Modelo Relacional Con base al modelo lógico de datos, se aplican las reglas para pasar al modelo relacional o diseño físico, en otras palabras, las entidades, relaciones y atributos se pasan a tablas y campos.  d) Implementación Una vez se tenga definida todas las tablas, con sus respectivos campos, llaves principales y llaves foráneas, se entra a definir en la herramienta de MySql. Lo anterior se realiza con las sentencias DDL (Lenguaje de Definición de datos).  **Definición de modelo y modelo de datos.**  Se entiende por modelo, el proceso de abstraer una realidad y tratar de modelarla a través de un gráfico, fórmula matemática, prototipo, etc. Para nuestro caso, entenderemos como modelo de datos, la abstracción del mundo real de la organización en término de datos y sus relaciones, plasmada en un gráfico llamado diagrama entidad-relación. Este diagrama, es una técnica de modelado conceptual (lógico) de datos, cuya base teórica se encuentran en la teoría de conjuntos, relaciones y funciones de la ciencia de las matemáticas. Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información. Con respecto al diseño de bases de datos, el modelado de datos puede ser escrito así (Brodie 1984:20): "dados los requerimientos de información y proceso en una aplicación de uso intensivo de datos (por ejemplo, un sistema de información), construir una representación de la aplicación que capture las propiedades estáticas y dinámicas requeridas para dar soporte a los procesos deseados (por ejemplo, transacciones y consultas). Además de capturar las necesidades dadas en el momento de la etapa de diseño, la representación debe ser capaz de dar cabida a eventuales futuros requerimientos". Un modelo de datos es por tanto una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. Conceptualmente, una aplicación puede ser caracterizada por:   * Propiedades estáticas: entidades (u objetos), propiedades (o atributos) de esas entidades, y relaciones entre esas entidades. * Propiedades dinámicas: operaciones sobre entidades, sobre propiedades o relaciones entre operaciones. * Reglas de integridad sobre las entidades y las operaciones (por ejemplo, transacciones).   Así, un modelo de datos se distingue de otro por el tratamiento que da a estas tres categorías. El resultado de un modelado de datos es una representación que tiene dos componentes: las propiedades estáticas se definen en un esquema y las propiedades dinámicas se definen como especificaciones de transacciones, consultas e informes.  Un esquema consiste en una definición de todos los tipos de objetos de la aplicación, incluyendo sus atributos, relaciones y restricciones estáticas. Correspondientemente, existirá una reposición de información, la base de datos, que es una instancia del esquema. Un determinado tipo de procesos sólo necesita acceder a un subconjunto predeterminado de entidades definidas en un esquema, por lo que este tipo de procesos puede requerir sólo un subconjunto de las propiedades estáticas del esquema general. A este subconjunto de propiedades estáticas se le denomina sub esquema.  Una transacción consiste en diversas operaciones o acciones sobre las entidades de esquema o sub esquema. Una consulta se puede expresar como una expresión lógica sobre los objetos y relaciones definidos en el esquema; una consulta identifica un subconjunto de la base de datos. Las herramientas que se usan para realizar las operaciones de definición de las propiedades estáticas y dinámicas de la base de datos son los lenguajes de definición y manipulación de datos (DDL, DML), junto con los lenguajes de consulta (SQL).  **Modelo Entidad - Relación**  El modelo E-R (Entidad-Relación) es un modelo de datos conceptual de alto nivel y que se suele utilizar bastante en el diseño de bases de datos Relacional. Se basa en una percepción del mundo real que consiste en un conjunto de objetos básicos denominados entidades y relaciones, y se desarrolló para facilitar el diseño de bases de datos.  El modelo E-R crea un modelo de la realidad que se asimila a la realidad que queremos modelar, y lo hace de forma que es independiente de la implementación posterior, ofreciendo un alto nivel de abstracción, y siendo una herramienta gráfica fácil de comprender.  El resultado del modelado E-R es un diagrama E-R que representa una estructura lógica general de la base de datos.  **Definición de Entidad y notación**  Es una abstracción de un conjunto de cosas (objetos) del mundo real, las cuales tienen las mismas características y están sujeta a las mismas reglas. Una entidad válida, debe ser significativa para el alcance del análisis, debe tener más de una ocurrencia y cada ocurrencia debe ser ÚNICA e identificable. Ejemplos de entidad es un conjunto de personas, vehículos, computadores, oficinas, facturas, créditos, etc. cabe resaltar que muchos de estos ejemplos con conjunto arbitrarios, solo creados por la imaginación de la mente humana, con el fin de organizar por categorías las cosas que lo rodean y poder tener mejor comprensión del mundo. |

|  |
| --- |
| **Nombre de la sección que se creara en el OVI: FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE BASES DE DATOS RELACIONAL** |
| **2.1 Objetivo de la sección: (**Registre a continuación el objetivo que tiene esta sección**)** |
| .  Apropiar al estudiante del modelo entidad-relación necesario para lograr un buen diseño de una base de datos relacional. |
| **2.2 Recursos de consulta que usara en la sección: (**coloque el nombre del material que usara para crear los contenidos de la sección y el enlace de descarga de los mismos sean estos Texto, Imágenes, Audios o Vídeos**)** |
| **Texto: Verdana 12** |
| **2.3 Redacte un borrador del contenido de lectura en formato de texto que tendrá la sección: (**Sea este la presentación de la sección, el contenido o ambos; redacte un borrador del texto que publicara como contenido en la sección coloque un subtítulo para identificar si corresponde a la presentación de la sección o el contenido de lectura de la sección**)** |
| **Definición de modelo y modelo de datos**  Se entiende por modelo, el proceso de abstraer una realidad y tratar de modelarla a través de un gráfico, fórmula matemática, prototipo, etc. Para nuestro caso, entenderemos como modelo de datos, la abstracción del mundo real de la organización en término de datos y sus relaciones, plasmada en un gráfico llamado diagrama entidad-relación. Este diagrama, es una técnica de modelado conceptual (lógico) de datos, cuya base teórica se encuentran en la teoría de conjuntos, relaciones y funciones de la ciencia de las matemáticas.  Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información.  **Modelo Entidad - Relación**  El modelo E-R (Entidad-Relación) es un modelo de datos conceptual de alto nivel y que se suele utilizar bastante en el diseño de bases de datos Relacional. Se basa en una percepción del mundo real que consiste en un conjunto de objetos básicos denominados entidades y relaciones, y se desarrolló para facilitar el diseño de bases de datos.  El modelo E-R crea un modelo de la realidad que se asimila a la realidad que queremos modelar, y lo hace de forma que es independiente de la implementación posterior, ofreciendo un alto nivel de abstracción, y siendo una herramienta gráfica fácil de comprender.  El resultado del modelado E-R es un diagrama E-R que representa una estructura lógica general de la base de datos.  **Definición de Entidad y notación**  Es una abstracción de un conjunto de cosas (objetos) del mundo real, las cuales tienen las mismas características y están sujeta a las mismas reglas. Una entidad válida, debe ser significativa para el alcance del análisis, debe tener más de una ocurrencia y cada ocurrencia debe ser ÚNICA e identificable. Ejemplos de entidad es un conjunto de personas, vehículos, computadores, oficinas, facturas, créditos, etc. cabe resaltar que muchos de estos ejemplos con conjunto arbitrarios, solo creados por la imaginación de la mente humana, con el fin de organizar por categorías las cosas que lo rodean y poder tener mejor comprensión del mundo.  **Tipos de entidades:**  Entidad fuerte o fundamental: es una entidad que se identifica por si sola, es decir, una o varias características (atributos) le garantiza UNICIDAD, por consiguiente, no depende de otra entidad o entidades. Gráficamente, tenemos:    Entidad débil: es una entidad que no se puede identificar por si sola, es decir, no existe una o varias características de la entidad que le pueda garantizar UNICIDAD, por lo tanto, depende de otra entidad y su dependencia es fuerte. Es de anotar, que en estas entidades como mínimo, una de sus características (atributos), hace parte de su CLAVE PRIMARIA. Gráficamente, tenemos a la entidad Historia del empleado.    Entidad asociativa: es una entidad débil, pero, depende de DOS o mas entidades, con el fin de garantizar unicidad. Es de anotar, que esta es la única entidad que puede o no tener características propias (atributos). Gráficamente, tenemos a la entidad Detalles de Facturas.    Definición de Atributo  Es una abstracción de las características que poseen todas las instancias u ocurrencias de una entidad. Estas características deben cumplir con ciertas propiedades:   * Incluir toda la información necesaria sobre la entidad. * No deben colocarse atributos de otra entidad. * Cada atributo debe referirse a una sola característica de la entidad. * Cada atributo debe tomar sus valores independientes a los demás.   Dominio: es un conjunto de valores que un atributo puede tomar. Para establecer los dominios se debe tener unas reglas de aceptación por parte de los responsables de la información y además citar los documentos que sustente las reglas acordadas. Estos pueden ser de diferentes tipos.   * Listas: cuando se requiere que un atributo tome valores que no están comprendidos en un rango. * Rango: son los valores aceptables en un intervalo continuo.   Atributo llave: es el conjunto de uno o más atributos, los cuales distinguen como única cada instancia u ocurrencia de una entidad.  **Definición de Relación y notación**  Es la asociación entre dos o más instancias del mismo o diferente tipo de entidad. Éstas, son relaciones simétricas, es decir, en doble sentido, de tal forma, que si una o varias instancias u ocurrencia de la entidad A, está relacionada, con una o varias ocurrencias de la entidad B, también una o varias instancias u ocurrencia de la entidad B, está relacionada, con una o varias ocurrencias de la entidad A. Por otro lado, pueden existir relaciones entre una o varias instancias u ocurrencia de la entidad A, con una o varias instancias u ocurrencia de la entidad A, es decir, con ella misma, lo que da un subconjunto del conjunto de A, lo mismo puede ocurrir con B.  **Definición de Cardinalidad y notación**  Es el establecimiento de número de instancias u ocurrencias que se relacionan. Para el establecimiento de la cardinalidad en un sentido, obsérvese en la gráfica, que siempre parte del mínimo y llega al máximo. Veamos ahora, las cuatro formas cardinales que tienen las relaciones: 1) Relación Cero a Uno (0:1): Es la relación donde para Cero instancias u ocurrencias de A, hay máximo Una instancia u ocurrencia de B. La interpretación que tiene la relación con esta cardinalidad, es que puede existir instancias u ocurrencias en B, sin necesidad de que exista instancias y ocurrencias en A. 2) Relación Uno a Uno (1:1): Es la relación donde para Una instancias u ocurrencias de A, hay máximo Una instancia u ocurrencia de B. La interpretación que tiene la relación con esta cardinalidad, es que para que exista una instancias u ocurrencias en B, hay necesidad de que exista instancias y ocurrencias en A. 3) Relación Cero a Muchos (0:N): Es la relación donde para Cero instancias u ocurrencias de A, hay máximo Muchas instancia u ocurrencia de B. La interpretación que tiene la relación con esta cardinalidad, es que puede existir una o varias instancias u ocurrencias en B, sin necesidad de que existan instancias y ocurrencias en A. 4) Relación Uno a Mucho (1:N): Es la relación donde para Una instancias u ocurrencias de A, hay máximo Muchas instancia u ocurrencia de B. La interpretación que tiene la relación con esta cardinalidad, es que, para que exista una o varias instancias u ocurrencias en B, hay necesidad de que existan una instancias y ocurrencias en A.  Es importante aclarar, que la existencia de los mínimos, dependen de las políticas y normas que tiene la empresa, así por ejemplo, si la empresa determina que para incluir un empleado , debe existir el cargo para esto, entonces se hace necesario que la relación de de cargo hacia empleados, como mínimo debe existir un cargo para poder inscribir empleados (1 - N). Ahora miremos en forma gráfica como queda: |

|  |
| --- |
| **Nombre de la sección que se creara en el OVI: LENGUAJE ESTANDAR DE CONSULTA Y HERRAMIENTAS** |
| **2.1 Objetivo de la sección: (**Registre a continuación el objetivo que tiene esta sección**)** |
| Apropiar al estudiante con conocimientos del lenguaje estándar de consultas (SQL) y herramientas que faciliten su comprensión y posterior uso. |
| **2.2 Recursos de consulta que usara en la sección: (**coloque el nombre del material que usara para crear los contenidos de la sección y el enlace de descarga de los mismos sean estos Texto, Imágenes, Audios o Vídeos**)** |
| **Texto: Verdana 12** |
| **2.3 Redacte un borrador del contenido de lectura en formato de texto que tendrá la sección: (**Sea este la presentación de la sección, el contenido o ambos; redacte un borrador del texto que publicara como contenido en la sección coloque un subtítulo para identificar si corresponde a la presentación de la sección o el contenido de lectura de la sección**)** |
| **LENGUAJE DE DEFINICIÓN DE DATOS**  El lenguaje de Definición de Datos, es el encargado de DEFINIR toda la estructura de una base de datos relacional. Para esto se apoya en tres instrucciones básicas como son el CREATE, el ALTER y el DROP. Los elementos de las bases de datos que se le puede aplicar estas instrucciones son bases de datos (DATABASE), tablas (TABLE), índices (INDEX) y vistas (VIEW). Miremos esquemáticamente lo anterior:    Obsérvese que las bases de datos, índices y vistas solo pueden ser creadas (CREATE) o borradas (DROP), mientras que las tablas, pueden también, ser modificadas, mediante la instrucción ALTER.  **Creación de Bases de Datos y Tablas**  La convención que se va a utilizar para cualquier instrucción es que las PALABRAS RESERVADAS se colocan en letra azul. Ahora, miremos en forma general como sería la instrucción de creación de cualquier elemento:  NOMB\_INSTRUCCION ELEMENTO Nomb\_Elemento  Tomando el ejemplo de Reservas de Hoteles, que se colocó al principio de esta unidad. Se procede así:  1. Creación de una base de datos Para el ejemplo llamaremos a la bases de datos Reservas\_Hoteles  CREATE DATABASE Reservas\_Hoteles;  2. Creación de una tabla  CREATE TABLE Municipios  (Cod\_Mun INT NOT NULL,  Nombre\_Municipio CHAR(30) NOT NULL,  PRIMARY KEY (Cod\_Mun));  CREATE TABLE Hoteles (Cod\_Hot CHAR(4) NOT NULL,  Hnombre CHAR(30) NOT NULL,  Cod\_Mun INT,  PRIMARY KEY (Cod\_Hot),  FOREIGN KEY (Cod\_Mun) REFERENCES Municipios (Cod\_Mun));  Haciendo un análisis hasta aqui, se puede observar que cuando finaliza una sentencia, se coloca punto y como (;). Claro que muchos motores de bases de datos, ya no ponen problema en esto, sino que si la sentencia están bien escrita la ejecuta. Otro punto es que en la definición de la llave foránea, hubo necesidad de Cualificar el campo, debido a que el nombre recibido por código municipio es igual en ambas tablas (Municipios y Hoteles).  CREATE TABLE Huespedes  (Nro\_Id\_Hues INT NOT NULL,  Nombre\_Hues CHAR(50) NOT NULL,  PRIMARY KEY (Nro\_Id\_Hues));  CREATE TABLE Reservas (Rnum INT NOT NULL,  Fecha\_Ini DATE,  Dias\_Duracion INT,  Cod\_Hot CHAR(4),  Nro\_Id\_Hues INT,  Nro\_Hab CHAR(4),  Fecha\_Res DATE,  PRIMARY KEY (Rnum),  FOREIGN KEY (Cod\_Hot) REFERENCES Hoteles (Cod\_Hot),  FOREIGN KEY (Nro\_Id\_Hues) REFERENCES Huespedes (Nro\_Id\_Hues));  Una aspecto a tener en cuenta con la definición de las llaves foráneas, es que la columna a la que se hace referencia, debe llamarse igual como se haya definido en la Tabla., así se observa que, Nro\_Id\_Hues que se encuentra entre paréntesis se llama exactamente igual a como se definió en la tabla de Huespedes. además obsérvese que se cualificó, debido a que tiene el mismo nombre en la tabla de Reservas y por lo tanto se debe evitarse las ambigüedades.  **Creación de Índices**  Los índices son los caminos que al motor de las bases de datos se le dan para que los encuentre de forma mas eficiente los datos, de tal forma, que escoja el camino más rápido para dar respuesta a una petición de un usuario. Un criterio fundamental para diseñar los índices, es la frecuencia de consulta de un dato o grupo de datos en una tabla y que no es llave primaria y el tamaño de la tabla. Sin embargo, hay que tener en cuenta que si bien se gana rapidez en la consulta, mientras más índices tenga una tabla, los procesos de inserción, borrado y modificación son mas demorado. De allí, que el administrador de bases de datos requiera mirar la forma como se ejecutó una consulta, una vez que crea un índice, con el fin de observar si el índice fue utilizado por el gestor de consulta, pues si este no fue utilizado, debe ser borrado, ya que no tiene sentido de acuerdo a lo dicho al inicio de este párrafo.  Ej:  CREATE INDEX Nombre\_Indice ON Nomb\_Tabla(Nombre\_Campos1, Nombre\_Campo2, ....);  CREATE INDEX Huesped ON Reservas(Nro\_Id\_Hues);  **Modificación de Tablas**  Ahora se muestra las sentencias para realizar tres tipos de modificaciones:  1. Adicionar la ciudad a la tabla de HUESPEDES.  ALTER TABLE HUESPEDES ADD Ciudad CHAR (30);  2. Borrar el nùmero de habitaciòn a las RESERVAS.  ALTER TABLE RESERVAS DELL Nto\_hab;  3. Modificar el tipo de datos dias de duraciòn en las RESERVAS.  ALTER TABLE RESERVAS MOD Dias\_Duracion CHAR (15);  **Borrados de Bases de Datos, Tablas, Índices y Vistas**  Para el borrado de cualquier elemento de la estructura de una base de datos, incluyendo ella misma, existe la sentencia DROP.  1. Borrado de una base de datos DROP DATABASE RESERVAS\_HOTELES;  2. Borrado de una tabla DROP TABLE MUNICIPIOS;  3. Borrado de un índice DROP INDEX HUESPEDES\_Nro\_Id\_Hues;  4. Borrado de una vista DROP VIEW VISTA\_RESERVAS;  **LENGUAJE DE MANIPULACIÓN DE DATOS**  La operaciòn de inserciòn, es la que permite crear los datos en las bases de datos. Para garantizar el èxito de esta operaciòn, es necesarios tener en cuentas estas situaciones:  1. La sentencia, solo permite insertar datos en UNA sola tabla.  2. Como mìnimo, se debe insertar los datos de los campos que conforman la LLAVE PRIMARIA.  3. Si existen campos, que no permiten valores NULOS, estos deben necesariamente, insertales valores.  4. Si la tabla, donde se van a insertar los datos, contiene LLAVES FORÀNEAS, debe garantizarse que al introducir un datos, este ya este creado en la tabla padre. Por lo tanto, es conveniente que al realizar inserciones en este tipo de tablas, lo primero que se debe hacer es insertar en las tablas que no tienen LLAVES FORANEAS, es decir, que son solo padres.  Cuando se insertan datos a una base de datos, se puede hacer de dos formas:  1. Insertando datos Fila por Fila  Se inserta la primera fila de la tabla de MUNICIPIOS y la segunda fila de la tabla de HOTELES.  INSERT INTO MUNICIPIOS (Cod\_Mun,Nombre\_municipio) VALUES(01,‘Bogotà');  INSERT INTO HOTELES (Cod\_Hot,Hnombre,Cod\_Mun) VALUES('H02,‘Hilton',01);  2. Insertando datos en varias filas  Cuando se utiliza la sentencia INSERT para agregar datos en varias filas, necesariamente se debe tomar con base en otra tabla, que tenga los campos estructurados de la misma forma de la tabla donde se va a insertar. Un ejemplo sería cargar una tabla temporal solo con los datos de número de reservas, número del huésped y fecha de reservas, cuya fecha inicial de reserva mayor a julio de 2011.  Lo anterior quedaría así:  INSERT INTO Temporal (Rnum, Nro\_Id\_Hues, Fecha\_Ini)  SELECT Rnum, Nro\_Id\_Hues, Fecha\_Ini  FROM RESERVAS  WHERE Fecha\_Ini > ‘01/07/2011‘;  Es de aclarar, que la tabla Temporal, se ha debido crear previamente, con la sentencia CREATE.  **Actualización de Filas**  La sentencia para actualizar se llama UPDATE. Cuando se ejecuta esta sentencia en su mínima expresión, se actualizan todas las filas. Obsérvese el siguiente ejemplo:  Actualizar el nombre del hotel ―Caribian‖, por ―Palacé‖  UPDATE HOTELES SET Hnombre = ‘Palacé’ WHERE Hnombre = ‘Caribian’:  Aquí es conveniente hacer varias precisiones. La cláusula WHERE, es opcional; si esta cláusula no se coloca, entonces todas las filas quedan con nombre “Palacé”, cosa que no es común que ocurra. En cambio, con la cláusula WHERE, solo cambia el que tiene el nombre de ‗Caribian‘, siendo así las cosas, podemos decir que la cláusula WHERE en la práctica, no es tan opcional.  **Borrado de Filas**  La cláusula para borrar filas es DELETE. Cuando se ejecuta esta sentencia en su mínima expresión, se borran todas las filas. Obsérvese el siguiente ejemplo: Vamos a borrar todas las filas de la tabla de reservas cuya fecha de inicio son del año 2010.  DELETE FROM HOTELES WHERE CIUDAD = ‘Armenia’;  Aquí vale la pena hacer la misma aclaración que hicimos con UPDATE, y es que la cláusula WHERE es opcional. De tal forma, que si no se coloca, entonces BORRA TODOS LOS DATOS DE LA TABLA, pero no la estructura.  **Consultas Basicas**  Para mirar toda la potencialidad de la sentencia SELECT, hemos organizado las consultas de menor a mayor grado de dificultad, por cuestiones pedagógicas. Comenzaremos con consultas sobre una sola tabla hasta finalizar con subconsultas, donde intervienen más de una tabla. Miremos la sentencia general  SELECT Campo1[,Campo 2, …..]  FROM Tabla 1[,Tabla 2, ……]  WHERE Condiciòn 1 [AND Condiciòn 2][OR Condiciòn 2][…….]  ORDER BY Campo 1 [,Campo 2, ….]  GROUP BY Campo 1 [,Campo 2, ….]  HAVING Condiciòn de Grupo [AND Condiciòn 2][OR Condiciòn 2][…….]  1. Consultas simples  a. Consulta de un determinado campo. Consultar el código, nombre y código municipios de todos los hoteles.  SELECT Cod\_Hot, Hnombre, Cod\_Mun FROM HOTELES;  b. Consultas de todos los campos. Consultar todos los datos de los hoteles.  SELECT \* FROM HOTELES;  c. Consultas con DISTINCT. Consultar el código de los hoteles que han tenido reservas.  SELECT DISTINCT Cod\_Hot FROM RESERVAS; Si no se coloca la palabra DISTINCT, saldría dos veces el código H01.  d. Consultas de valores calculados. Consultar el nombre de los hoteles con sus respectivas tarifas, aumentadas en un 10%.  SELECT Hnombre, ―Tarifa Aumentada en 10%‖,Tarifa\_Noche\*1.1 FROM HOTELES;  e. Consultas con campos cualificados. Consultar el código y el nombre de los municipios.  SELECT MUNICIPIOS.Cod\_Mun,MUNICIPIOS.Nomb\_Mun  FROM MUNICIPIOS;  f. Consultas con condición. Consultar el código del hotel que han reservados en el año 2011  SELECT DISTINCT Cod\_Hot FROM RESERVAS WHERE Fecha\_Ini=‘01/01/2011‘ AND Fecha\_Ini=‘31/12/2011‘;  g. Consultas con ordenamiento. Consultar el código del hotel y la fecha de reservas, organizado por código hotel.  SELECT DISTINCT Cod\_Hot, Fec\_Res FROM RESERVAS ORDER BY Cod\_Hot;  2. Consultas de Reunión  a. Reunión simple. Consultar el nombre de los hoteles que han tenido reservas  SELECT DISTINCT Hnombre FROM HOTELES,RESERVAS WHERE HOTELES.Cod\_Hot=RESERVAS.Cod\_Res;  3. Consultas con Funciones Agregadas.  COUNT(\*): Cuenta los registros de una tabla  COUNTO(Campo): Cuenta los registros por un campo determinado. Si el campo tiene un valor nulo, no lo cuenta.  SUM(Campo): Suma todos los valores del campo. No suma nulos. el campo debe ser numérico.  AVG(Campo): Promedia el campo. No tiene en cuenta los nulos y el campo debe ser numérico.  MAX(Campo): Selecciona el valor máximo del campo. No cuenta el valor nulo. MIN(Campo): Selecciona el valor mínimo del campo. No cuenta el valor nulo.  a. Consultas para contar filas. Consultar el número total de hoteles.  SELECT COUNT (\*) FROM HOTELES;  b. Consultas para contar final con condición. Obtener la cantidad total de hoteles que se encuentran en el municipio H01.  SELECT COUNT(Cod\_Hot) FROM HOTELES WHERE Cod\_Mun=‘H01‘ ;  c. Consultas con agrupación. Obtener el número total de reservas por hotel  SELECT Cod\_Hot,COUNT(Nro\_Res) FROM RESERVAS GROUP BY Cod\_Hot;  En la consulta anterior, cuando tenemos en la clausula SELECT campos acompañados con funciones agregadas, siempre debe agruparse por estos campos, o si no, sale error en la consulta y NO se ejecuta.  d. Consultas con agrupación y reunión de tablas. Consultar el nombre del hotel con su respectivo número total de reservas.  SELECT Hnombre,COUNT(Nro\_Res) FROM HOTELES,RESERVAS WHERE HOTELES.Cod\_Hot=RESERVAS.Cod\_Hot GROUP BY Hnombre;  En esta consulta, tenemos el mismo caso de la clausula SELECT. Campos acompañados con funciones agregadas, siempre debe agruparse por ellos, o si no, sale error en la consulta y NO se ejecuta.  e. Consultas con condición de agrupación. Consultar el nombre del hotel, con su respectiva cantidad total de días reservados; solos aquellos hoteles que tienen más de dos días reservados.  SELECT Hnombre,SUM(Dias\_Dur) FROM HOTELES,RESERVAS WHERE HOTELES.Cod\_Hot=RESERVAS.Cod\_Hot GROUP BY Hnombre HAVING SUM(Dias\_Dur)>2; .  En esta consulta, tenemos el mismo caso de la cláusula SELECT. Campos acompañados con funciones agregadas, siempre debe agruparse por ellos, o si no, sale error en la consulta y NO se ejecuta. Aquí la cláusula HAVING se ejecuta después de que el motor ha hecho el agrupamiento y sumado los días de duración, es decir, esta condición es de grupo, no por fila como la WHERE .  f. Consultas con condición y agrupamiento varios. Consultar el nombre del hotel, y el nombre de los municipios, con su respectiva cantidad de reservas. Solo los hoteles del municipio de Cartagena o Bogotá, y que tengan más de 2 reservas.  SELECT Hnombre,Nomb\_Mun,SUM(Dias\_Dur)  FROM HOTELES,MUNICIPIOS,RESERVAS  WHERE HOTELES.Cod\_Hot=RESERVAS.Cod\_Hot  AND HOTELES.Cod\_Mun=MUNICIPIOS.Con\_Mun  AND (MUNICIPIOS.Nomb\_Mun=‘Cartagena‘ OR  MUNICIPIOS.Nomb\_Mun=‘Bogotá‘)  GROUP BY Hnombre, Nomb\_Mun  HAVING SUM(Dias\_Dur)>1;  **Consultas Avanzadas**  a. Consultas con el operador LIKE. Consultar los hoteles cuyo nombre empiezan con la letra ‗A‘.  SELECT \*  FROM HOTELES  WHERE Hnombre LIKE ‘A%‘ ;  b. Consultar los nombres de los hoteles cuya segunda letra sea „l‟.  SELECT \*  FROM HOTOLES  WHERE Hnombre LIKE ‘A%‘ ;  c. Consultar los nombres de los Municipios cuya última letra sea igual a “E‟.  SELECT \* FROM MUNICIPIOS  WHERE Nomb\_Mun LIKE ‘%E‘ ;  d. Consultar los proveedores que tengan en el nombre la letra “S‟.  SELECT \* FROM MUNICIPIOS WHERE Nomb\_Mun LIKE ‘%t%‘ ;  e. Subconsultas. Consultar los hoteles que no han tenido reservas.  SELECT \*  FROM HOTELES  WHERE Cod\_Hot NOT IN (SELECT Cod\_Hot FROM RESERVAS) ;  Nota: Esta consulta se puede realizar también utilizando el  EXISTS SELECT \*  FROM HOTELES  WHERE NOT EXISTS (SELECT Cod\_Hot FROM RESERVAS WHERE RESERVAS.Cod\_Hot=HOTELES. Cod\_Hot)  **HERRAMIENTA MySQL**  MySQL AB es una empresa comercial fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación. MySQL AB se estableció originalmente en Suecía por David Axmark, Allan Larsson, y Michael ―Monty‖' Widenius MySQL, el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. Es un sistema de gestión de bases de datos relacionales. Es software Open Source. El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar. Desarrollado originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes. Trabaja en entornos cliente/servidor o incrustados. |

BIBLIOGRAFIA

SÁNCHEZ, M. (2011). 301330 –Bases de Datos Básicos. Recuperado de <http://www.unad.edu.co>