

Universidad Nacional Autónoma de México

Facultad de Ingeniería

**Proyecto Final**

Bases de Datos

Grupo 2

Profesora: López Pelcastre Martha

Integrantes:

Cruz Zavala Miguel Ángel

Domínguez Cisneros Alexis Saúl

Monroy Velázquez Alejandra Sarahí

Semestre 2020-1

15 de Noviembre de 2019

**ENUNCIADO**

Se requiere diseñar una base de datos para un concesionario de vehículos de primera y segunda mano. El Comité de Dirección de la empresa quiere que se registre la actividad acerca de las ventas de vehículos, las reparaciones de los vehículos y de sus revisiones. En una entrevista con el personal de la concesionaria se obtuvo la siguiente información.

Registro y modificación de ventas. Sobre una venta, se ha de poder identificar el coche vendido, el tipo, el modelo, el cliente que ha adquirido el coche, la fecha en la que se ha vendido el coche, el comercial responsable de la venta, extras de la venta (pintura especial, todo eléctrico, etcétera) y el costo total de la venta. Un cliente ha de poder comprar más de un coche. Una venta debe tener forzosamente el auto vendido, quien lo vendó y el monto.

Para un cliente comprador, se ha de poder registrar su nombre, teléfonos de contacto e e-mails, identificador, dirección (calle, número, colonia, municipio, estado), sexo (F,M) y fecha de nacimiento. Por lo menos debe existir un dato de contacto para cada cliente. La base de datos debe notificar si el cliente ya existe, el cual puede ser buscado por cualquier parte de su nombre o RFC.

Se ha de poder identificar los comerciales en el sistema y han de ser usuarios permitiéndoles registrar sus propias ventas.

Para cada venta se permite modificar la fecha/hora, el comercial responsable, el modelo del coche, junto con sus extras o características y el cliente que ha comprado el coche.

Para cada reparación o revisión se debe poder gestionar la información identificativa del cliente, el coche afectado, la fecha y hora, el motivo de la reparación/revisión, el tiempo que se ha tardado en realizar la reparación o revisión, el mecánico responsable y costo de las mismas(desglosar el iva). De las revisiones se ha de poder saber la fecha de planificación y si ya se ha realizado o no. En caso afirmativo la fecha en la que se realizó.

Se debe de poder definir un catálogo con los diferentes tipos de reparaciones (cambio de parabrisas, caja de velocidad, etc.) y revisiones (cambio de aceite, afinación, servicio 50,000 km).

Cada reparación/revisión tiene un estatus, los estatus son (en proceso, concluido, entregado). Los técnicos pueden hacer revisiones, pero no reparaciones.

Se anula la transacción en los casos siguientes:

a) el coche referenciado no existe

b) el técnico referenciado no existe

c) no se ha especificado una fecha de realización ni una fecha de planificación

d) si se ha especificado una fecha de realización, pero no se ha especificado el mecánico o no se ha especificado un valor para la duración

e) si se ha especificado una fecha de planificación y los campos duración o técnico no contienen algún valor.

Datos sobre los coches: modelo, matrícula, extras/características. Propietario, costo y fecha para el caso de los autos de segunda mano. Tomar en cuenta que solo se puede comprar 3 autos a un mismo propietario y que éstos pueden comprar o reparar autos (llevar sus autos a reparación).

Se tiene un listado de los extras que pueden ser agregados a los autos nuevos para su venta.

La base de datos debe notificar si la matricula la existe para no registrar duplicados.

De los comerciales, mecánicos y técnicos se desea almacenar; nombre completo, fecha de ingreso número de empleado, fecha de nacimiento y sueldo. Datos de su domicilio, teléfonos y e-mails. A los comerciales se les da una comisión por cada venta. De los mecánicos se desea saber escolaridad y cursos tomados.

A partir del registro de esta información, interesa poder obtener estadísticas de por ejemplo; qué comerciales realizan más ventas en un periodo de tiempo, en qué épocas se realizan más ventas, qué vehículos son los más vendidos, qué mecánicos realizan el mayor número de revisiones y reparaciones. Realice los procedimientos para poder obtener estas 4 estadísticas. Dichos procedimientos deben mostrar al menos 20 registros para cada caso.

Debe usar vistas por lo menos 3.

La dirección ha decidido recientemente que la empresa debe mejorar en el registro de la actividad de la empresa con el fin de poder analizar la actividad en más detalle. Se espera que el análisis más exhaustivo de la actividad permita facilitar la toma de decisiones estratégicas futuras.

Utilizar esquemas para la base de datos al menos 3

Crear los scripts para crear los usuarios a las bases de datos que reciba como parámetro el usuario y el password. Considerar que habrá usuarios de solo consulta, otros que agreguen o actualicen información y los administradores. Por defecto elaborar el script para crear los siguientes usuarios usuarioConsulta, usuarioConcesionario (administrador), con password 1234zaq\*

**Diseño Conceptual**

*Identificación de sustantivos, entidades y relaciones para la creación del modelo*

|  |  |  |
| --- | --- | --- |
| Sustantivos | Entidades | Relaciones |
| modelo  matricula  extras  propietario  costo  fecha  fechaAjuste  hora  motivo  estatus  fechaPlanificacion  idAjuste  iva  nombreCliente  apPatCliente  apMatCliente  telefonoCliente  e-mail  idCliente  calle  numero  colonia  municipio  estado  sexo  fechaNaCliente  nombreEmpleado  apPatEmpleado  apMatEmpleado  fechaIngreso  numeroEmpleado  fechaNacEmp  sueldo  calleEmpleado  numeroEmpleado  coloniaEmpleado  municipioEmpleado  estadoEmpleado  telefonoEmpleado  e-mailEmpleado  comision  cursos  escolaridad  costoTotal  fechaCompra  extras  idCompra | COMPRA  CLIENTE  COCHE   * USADO * NUEVO   AJUSTE   * REPARACION * REVISION   EMPLEADO   * MECANICO * TECNICO * COMERCIAL | Un cliente puede comprar más de un coche  Un técnico puede hacer revisiones, pero no reparaciones  Un mecanico puede hacer tanto reparaciones como revisiones  Usados solo se pueden comprar 3 autos a un mismo propietario y los propietarios pueden comprar y mandas a reparar autos  Un comercial debe realizar una venta |

*Imagen que contiene texto, mapa

Descripción generada automáticamenteModelo ER en día:*

**DISEÑO LOGICO**

*Esquema Relacional*

COCHE = { matricula (PK), modelo, costo, tipoCoche (CK) }

* tipoCoche es un check que verifica si un coche es nuevo o usado.

USADO = { matricula (PK, FK), idCliente (FK) }

EXTRASCOCHE = { matricula (PK, FK), descripcionExtra (PK) }

CLIENTE = { idCliente (PK), sexo (CK), fechaNacCliente, rfc (U), nombreCliente, apPatCliente, apMatCliente (N), calle, numero, colonia, municipio, estado }

* sexo solo puede ser F o M.

EMAIL = { idCliente (PK, FK), email (PK) }

TELEFONO = { idCliente (PK, FK), teléfono (PK) }

EMPLEADO = { numEmpleado (PK), nombreEmpleado, apPatEmpleado, apMatEmpleado (N), fechaIngreso, fechaNacEmpleado, sueldo, calleEmpleado, numeroEmpleado, coloniaEmpleado, municipioEmpleado, estadoEmpleado, tipoEmpleado (CK) }

* tipoEmpleado verifica si el empleado es comercial, técnico o mecánico.

EMAILEMPLEADO = { numEmpleado (PK, FK), emailEmpleado (PK) }

TELEFONOEMPLEADO = { numEmpleado (PK, FK), telefonoEmpleado (PK) }

COMERCIAL = { numEmpleado (PK, FK) }

TECNICO = { numEmpleado (PK, FK) }

MECANICO = { numEmpleado (PK, FK), escolaridad }

CURSO = { numEmpleado (PK, FK), curso (PK) }

COMPRA = { idCompra (PK), fechaCompra, costoTotal, idCliente (FK), matricula (FK) }

REGISTRACOMPRA = { [ idCompra (FK), numEmpleado (FK) ] (PK), comisión }

AJUSTE = { [ matricula (FK), idCliente (FK), numEmpleado (FK), idAjuste(D) ] (PK), costoAjuste, fechaAjuste, duración, iva(C), estatus (CK), tipoAjuste (CK) }

* El iva es calculado a partir del costo del ajuste.
* estatus verifica si esta en proceso, concluido o entregado.
* tipoAjuste verifica si es revisión o reparación.

REVISION = { [ matricula (FK), idCliente (FK), numEmpleado (FK), idAjuste(D) ] (PK, FK), fechaPlanificacion, fechaPlanificacionRealizada (N), checkPlanRealizada(CH)}

* checkPlanRealizada es un check que verifica si la fechaPlanificación ya se ha realizado

*Modelo Lógico en ERwin*

En este modelo ya se han agregado los catálogos TIPOREVISION Y TIPOREPARACION, además de agregar las nuevas llaves artificiales en el caso de EXTRASCOCHE, TELEFONO, EMAIL, TELEFONOEMPLEaADO, EMAILEMPLEADO, y AJUSTE, esta ultima le hereda su llave artificial a REPARACION Y REVISION. Quedando el modelo lógico final como:

*Normalización*

Nuestro diseño conceptual nos garantizó que nuestra base de datos estuviera normalizada como mínimo hasta la tercera forma normal debido que al hacer nuevamente el análisis a nuestro modelo lógico tenemos que:

* *1era Forma Normal:* Cada uno de nuestros atributos en cada relación son atómicos, es decir son indivisibles, como por ejemplo en el caso de los nombres o direcciones, ya están divididos por los campos que constituyen a cada uno; tampoco tenemos valores repetidos y cada relación cuenta con una clave primaria propia.
* *2da Forma Normal:* Cumple con la primera forma normal y además cada uno de los atributos dependen de la clave primaria de cada relación.
* *3era Forma Normal:* Cumple con la segunda forma normal y en el caso de cuando nuestras relaciones tienen claves foráneas que no son primarias, no existe una dependencia entre estas claves y los atributos de dicha relación. Por lo que también cumple con tercera forma normal.

*Diccionario de Datos*

|  |  |  |
| --- | --- | --- |
| ***Nombre*** | ***Tipo*** | ***Descripción*** |
| COCHE | Tabla | Contiene los atributos de coche. |
| EXTRASCOCHE | Tabla | Contiene los extras que podría tener un coche |
| USADO | Tabla | Contiene el id del cliente propietario que vende el coche de segunda mano. |
| CLIENTE | Tabla | Contiene todos los datos de los clientes del concesionario. |
| TELEFONO | Tabla | Contiene el campo para guardar los teléfonos del cliente. |
| EMAIL | Tabla | Contiene el campo para guardar los emails del cliente. |
| EMPLEADO | Tabla | Contiene todos los datos de los empleados. |
| COMERCIAL | Tabla | Contiene el id del empleado ya que este se relacionara con la tabla REGISTRACOMPRA. |
| TECNICO | Tabla | Contiene el id del empleado, ya que este se relacionara con la tabla REVISION. |
| MECANICO | Tabla | Contiene el id del empleado ya que este se relacionara con la tabla AJUSTE, además contiene el campo para guardar la escolaridad del mecánico. |
| CURSO | Tabla | Contiene el campo para guardar los cursos realizados por el mecánico. |
| TELEFONOEMPLEADO | Tabla | Contiene el campo para guardar los teléfonos del empleado. |
| EMAILEMPLEADO | Tabla | Contiene el campo para guardar los emails del empleado. |
| COMPRA | Tabla | Contiene los campos para almacenar los datos necesarios para registrar una compra. |
| REGISTRACOMPRA | Tabla | Contiene los campos necesarios para almacenar la compra y el empleado que registra la compra, además de la comisión generada por esa compra. |
| AJUSTE | Tabla | Contiene los campos para registrar un ajuste. |
| REVISION | Tabla | Contiene los campos para registrar una revisión. |
| TIPOREVISION | Tabla | Contiene la descripción de la revisión. |
| REPARACION | Tabla | Contiene los campos para registrar una reparación. |
| TIPOREPARACION | Tabla | Contiene la descripción de la reparación. |
|  |  |  |
| ***Nombre del atributo*** | ***Tipo de dato*** | ***Longitud*** |
| matricula | char | (7) |
| modelo | varchar | (20) |
| costo | decimal | (10,2) |
| tipoCoche | char | (5) |
| idExtraCoche | char | (7) |
| descripcionExtra | char | (20) |
| idCliente | integer |  |
| rfc | char | (18) |
| sexo | char | (1) |
| fechaNacCliente | date |  |
| nombreCliente | varchar | (30) |
| apPatCliente | varchar | (30) |
| apMatCliente | varchar | (30) |
| calle | varchar | (20) |
| numero | varchar | (20) |
| colonia | varchar | (20) |
| municipio | varchar | (30) |
| estado | varchar | (30) |
| idEmail | integer |  |
| email | varchar | (30) |
| idtelefono | char | (10) |
| idCompra | integer |  |
| fechaCompra | date |  |
| costoTotal | integer |  |
| numEmpleado | integer |  |
| nombreEmpleado | varchar | (30) |
| apPatEmpleado | varchar | (30) |
| apMatEmpleado | varchar | (30) |
| fechaIngreso | date |  |
| fechaNacEmpleado | date |  |
| sueldo | decimal | (10,2) |
| calleEmpleado | varchar | (20) |
| numeroEmpleado | varchar | (20) |
| coloniaEmpleado | varchar | (20) |
| municipioEmpleado | varchar | (20) |
| estadoEmpleado | varchar | (20) |
| tipoEmpleado | varchar | (10) |
| idEmailEmpleado | integer |  |
| emailEmpleado | varchar | (30) |
| idTelefonoEmpleado | integer |  |
| telefonoEmpleado | char | (10) |
| escolaridad | varchar | (20) |
| idCurso | integer |  |
| curso | varchar | (20) |
| artIdAjuste | integer |  |
| idAjuste | integer |  |
| costoAjuste | decimal | (10,2) |
| fechaAjuste | date |  |
| duracion | varchar | (15) |
| iva | decimal | (8,2) |
| estatus | varchar | (20) |
| tipoAjuste | varchar | (10) |
| fechaPlanificacionRealizada | date |  |
| fechaPlanificacion | date |  |
| checkPlanRealizada | char | (1) |
| idTipoRevision | integer |  |
| descripcionRev | varchar | (20) |
| idTipoReparacion | integer |  |
| descripciónRep | varchar | (20) |
|  |  |  |
| ***Nombre Índice*** | ***Tipo de Índice*** | ***Campo*** |
| PK\_ajuste | Clustered | artIdAjuste |
| PK\_cliente | Clustered | idCliente |
| PK\_coche | Clustered | matricula |
| PK\_comercial | Clustered | numEmpleado |
| PK\_compra | Clustered | idCompra |
| PK\_empleado | Clustered | numEmpleado |
| PK\_mecanico | Clustered | numEmpleado |
| PK\_reparacion | Clustered | artIdAjuste |
| PK\_revision | Clustered | artIdAjuste |
| PK\_tecnico | Clustered | numEmpleado |

|  |  |
| --- | --- |
| ***Nombre de la Función*** | ***Descripción*** |
| fu\_mecanicosNoRegistrados() | Función que nos regresa una tabla con los mecánicos existentes en la tabla EMPLEADO pero que no han sido añadidos a la tabla MECANICO.  En el caso de que todos los empleados de tipo mecánico que existen en la tabla EMPLEADO ya existan también en la tabla MECANICO nos regresa una tabla con un único registro con la cadena 'Ninguno'.  La columna de dicha tabla recibe el nombre de 'registrarNumeroDeEmpleadoEnTablaMecanico' que quiere decir que se debe introducir  en la tabla MECANICO un registro cuyo numEmpleado sea igual a el valor en cada registro de tal columna |
| fu\_usadosNoRegistrados() | Función que nos regresa una tabla con los coches de tipo 'usado' existentes en la tabla COCHE pero que no han sido añadidos a la tabla USADO.  En el caso de que todos los coches de tipo 'usado' que existen en la tabla COCHE ya existan también en la tabla USADO nos regresa una tabla con un único registro con la cadena 'Ninguno'.  La columna de dicha tabla recibe el nombre de 'registrarMatriculaDeCocheEnTablaUsado' que quiere decir que se debe introducir  en la tabla USADO un registro cuya matrícula sea igual a el valor en cada registro de tal columna. |

|  |  |
| --- | --- |
| ***Nombre del Trigger*** | ***Descripción*** |
| TR\_herenciaEmpleado | Trigger que hace que sea posible la inserción múltiple y verifica los diferentes tipos de  empleados que son insertados, según sea el tipo de empleado hacemos cosas diferentes.  Si el tipo de empleado es un comercial o un técnico, se usa el número de empleado que se usó para  insertar en la tabla EMPLEADO para insertar en las tablas respectivas de COMERCIAL o TECNICO según sea el caso.  Si el tipo de empleado es un mecánico este no es insertado automáticamente en la tabla MECANICO.  Gracias al uso de un cursor podemos hacer todo lo anterior incluso con inserciones múltiples, ya que  usa la tabla inserted para corroborar cada uno de los pasos anteriores con cada valor insertado.  Al final nos muestra una tabla la cual nos dice cuales mecánicos debemos registrar manualmente en la tabla MECANICO. |
| TR\_mecanicosRestantes | El trigger funciona con inserts múltiples gracias al uso de un cursor y la tabla INSERTED.  Para cada dato insertado se verifica que el numEmpleado exista en la tabla EMPLEADO y que además sea  un empleado de tipo 'Mecánico', si cumple esas restricciones los valores se registran en la tabla MECANICO.  En el caso contrario se imprime un mensaje que especifica que el empleado no existe en la tabla. |
| TR\_confirmarReparacion | Trigger que se encarga de verificar que el ajuste exista en la tabla AJUSTE y además que dicho ajuste sea de tipo 'Reparación', si lo es se inserta la reparación, en caso contrario  se manda un mensaje para verificar la existencia del ajuste en la tabla ajuste y que sea de tipo  'Reparación'. Este trigger no está diseñado para inserts múltiples. |
| TR\_confirmarRevision | Trigger que se encarga de verificar que el ajuste exista en la tabla AJUSTE y además que dicho ajuste sea de tipo 'Revisión', si lo es se inserta la revisión, en caso contrario se manda un mensaje para verificar la existencia del ajuste en la tabla ajuste y que sea de tipo  'Revisión'. Este trigger no está diseñado para inserts múltiples. |
| TR\_herenciaCoche | Trigger que nos mostrara, en el caso de que hayamos registrado al menos un coche de tipo usado, la matricula  de los coches usados que se deben insertar manualmente en la tabla USADO. |
| TR\_usadosRestantes | Trigger que después de insertar en USADO verifica que sea de tipo usado antes de insertar y que exista dicha matricula en la tabla COCHE para finalmente checar que no falten más coches usados que hayan sido registrados en la tabla COCHE sin registrar en la tabla USADO. |
| TR\_contactoCliente | Trigger para avisar que es necesario insertar un dato de contacto para el/los clientes insertados. |

|  |  |
| --- | --- |
| ***Nombre de la vista*** | ***Descripción*** |
| vsMECANICO | Vista que muestra el número de empleado y nombre completo de los mecánicos. |
| vsCOMERCIAL | Vista que muestra el número de empleado y nombre completo de los comerciales. |
| vsUSADO | Vista que muestra la matricula, modelo y costo de los coches usados. |

|  |  |
| --- | --- |
| ***Nombre del Procedimiento*** | ***Descripción*** |
| spu\_creaUsuarios | Procedimiento que recibe usuario, contraseña y tipo de usuario que se quiere crear, regresa el usuario creado con los privilegios correspondientes a cada tipo de usuario: administrador, consulta y modificaciones. |
| spu\_mejorComercial | Procedimiento que recibe dos parámetros correspondientes a un periodo de tiempo, regresa el top 5 de los comerciales que han hecho más ventas en ese periodo, las enumera y las ordena de forma descendente. |
| spu\_epocaVentas | Procedimiento que regresa las estaciones y cuantas ventas se registraron en cada estación. |
| spu\_cochesMasVendidos | Procedimiento que regresa el top 5 de los vehículos que más se han vendido, mostrando modelo y cantidad y los ordena de forma descendente. |
| spu\_mejorMecanico | Procedimiento que regresa el top 5 de los mecánicos que más revisiones/reparaciones han hecho y los ordena de forma descendente. |

**DISEÑO FISICO**

|  |  |  |
| --- | --- | --- |
| ***Punto*** | ***Nombre del Archivo*** | ***Observaciones*** |
| A) | seguridad.sql | Script que contiene un procedimiento almacenado el cual recibe usuario, contraseña y tipo de usuario que se quiere crear, regresa el usuario creado con los privilegios correspondientes a cada tipo de usuario: administrador, consulta y modificaciones. |
| B) | creaBase.sql | Script que contiene todas las tablas con sus respectivos campos necesarias para la base de datos, las relaciones entre ellas. También se incluyen los índices, valores por default, y checks. |
| C) | dml.sql | Script que contiene todas las funciones, triggers y vistas. |
| D) | informes.sql | Script que contiene los procedimientos almacenados para generar cada uno de los informes. Cada procedimiento utiliza una vista para realizar el proceso de la estadística correspondiente. |
| E) | cargaInicial.sql. | Script que contiene las inserciones de al menos 20 registros en cada tabla de la base de datos. |
| F) | validaTriggers.sql | Script que contiene cada una de las validaciones de los triggers que se crearon en la base de datos. |

*Consideraciones*

*Esquemas:*

- VENTA : Contiene a la tabla CLIENTE, COCHE, COMPRA, EMAIL, EXTRASCOCHE, REGISTRACOMPRA, TELEFONO y USADO.

- EMPLEADO : Contiene a la tabla COMERCIAL, CURSO, EMAILEMPLEADO, EMPLEADO, MECANICO, TECNICO y TELEFONOEMPLEADO.

- SERVICIO : Contiene a la tabla AJUSTE, REPARACION, REVISION, TIPOREPARACION y TIPOREVISION.

1.- *Restricciones* : A continuación, se lista el nombre de la tabla, y campo que contiene la restricción correspondiente.

AJUSTE : (artIdAjuste) – Primary Key

(matricula) – Foreign Key

(numEmpleado) – Foreign Key

(idCliente) – Foreign Key

(matricula, numEmpleado, idCliente, idAjuste) – Unique

(estatus='enProceso' or estatus='concluido' or estatus='entregado') – Check

(tipoAjuste='revision' or tipoAjuste='reparacion') – Check

CLIENTE : (idCliente) – Primary Key

(rfc) – Unique

(sexo='M' or sexo='F') – Check

COCHE : (matricula) – Primary Key

(tipoCoche='usado' or tipoCoche='nuevo') – Check

COMERCIAL : (numEmpleado) – Primary Key

(numEmpleado) – Foreign Key

COMPRA : (idCompra) – Primary Key

(idCliente) – Foreign Key

(matricula) – Foreign Key

CURSO : (idCurso) – Primary Key

(numEmpleado) – Foreign Key

(numEmpleado, curso) – Unique

EMAIL : (idEmail) – Primary Key

(idCliente) – Foreign Key

(idCliente, email) – Unique

EMAILEMPLEADO : (idEmailEmpleado) – Primary Key

(numEmpleado) – Foreign Key

(numEmpleado, emailEmpleado) – Unique

EMPLEADO : (numEmpleado) – Primary Key

(tipoEmpleado='Comercial' or tipoEmpleado='Mecanico' or tipoEmpleado='Tecnico') – Check

EXTRASCOCHE : (idExtraCoche) – Primary Key

(matricula) – Foreign Key

(matricula, descripcionExtra) – Unique

MECANICO : (numEmpleado) – Primary Key

(numEmpleado) – Foreign Key

REGISTRACOMPRA : (idCompra, numEmpleado) – Primary Key

(idCompra) – Foreign Key

(numEmpleado) – Foreign Key

REPARACION : (artIdAjuste) – Primary Key

(artIdAjuste) – Foreign Key

(idTipoReparacion) – Foreign Key

REVISION : (artIdAjuste) – Primary Key

(numEmpleado) – Foreign Key

(idTipoRevision) – Foreign Key

(checkPlanRealizada='S' or checkPlanRealizada='N') – Check

TECNICO : (numEmpleado) – Primary Key

(numEmpleado) – Foreign Key

TELEFONO : (idTelefono) – Primary Key

(idCliente) – Foreign Key

(idCliente, telefono) – Unique

TELEFONOEMPLEADO : (idTelefonoEmpleado) – Primary Key

(numEmpleado, telefonoEmpleado) – Unique

(numEmpleado) – Foreign Key

TIPOREPARACION : (idTipoReparacion) – Primary Key

TIPOREVISION : (idTipoRevision) – Primary Key

USADO: (matricula) – Primary Key

(matricula) – Foreign Key

(idCliente) – Foreign Key

2.- *Valores por default.*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabla*** | ***Campo*** | ***Valor por default*** | ***Descripción*** |
| EMPLEADO | sueldo | 4000 | Si no se recibe un valor en el campo sueldo del empleado, se le pondrá 4000 como valor de default. |
| REGISTRACOMPRA | comision | 1000 | Si no se recibe un valor en el campo comision del empleado, se le pondrá 1000 como valor de default. |
| REVISION | checkPlanRealizada | ‘N’ | Si no se recibe un valor en este campo, se le pondrá una ‘N’, que corresponde a que la fecha de planificación no se ha realizado. |

3.- *Llaves* *Artificiales*.

|  |  |  |
| --- | --- | --- |
| ***Tabla*** | ***Nombre de la llave Artificial*** | ***Descripción*** |
| EXTRASCOCHE | idExtraCoche | Asignamos la llave artificial a esta tabla y volvemos unique a matricula y descripcionExtra que originalmente eran la Primary Key. |
| TELEFONO | idTelefono | Asignamos la llave artificial a esta tabla y volvemos unique a telefono y idCliente que originalmente eran la Primary Key. |
| EMAIL | idEmail | Asignamos la llave artificial a esta tabla y volvemos unique a email y idCliente que originalmente eran la Primary Key. |
| TELEFONOEMPLEADO | idTelefonoEmpleado | Asignamos la llave artificial a esta tabla y volvemos unique a telefonoEmpleado y numEmpleado que originalmente eran la Primary Key. |
| EMAILEMPLEADO | idEmailEmpleado | Asignamos la llave artificial a esta tabla y volvemos unique a emailEmpleado y numEmpleado que originalmente eran la Primary Key. |
| AJUSTE | artIdAjuste | Asignamos la llave artificial a esta tabla y volvemos unique idAjuste, matricula, numEmpleado y idCliente, que originalmente eran la Primary Key. |

Nota: La llave artificial de Ajuste se hereda a Reparación y Revisión como FK y PK.

4.- *Consultas.(Joins, algebra relacional, funciones de agregación y subconsultas.):* Para este apartado se utilizaron subconsultas y consultas tantos en las funciones como en cada uno de los triggers implementados, estos se encuentran en el script dml.sql. Las funciones de agregación se utilizaron en las consultas implementadas en el script informes.sql, ahí también podemos observar que en las consultas se utilizaron los diferentes tipos de joins.

5.- *Estadísticas:* Los procedimientos almacenados para cada estadística se encuentran descritos en el diccionario de datos, en la sección de procedimientos almacenados, exceptuando el primer procedimiento mencionado en esa sección, los tres siguientes procedimientos corresponden a las estadísticas.

6.- *Uso de índices:* Los índices se encuentran descritos en el diccionario de datos en la sección de índices, donde se puede observar el nombre del índice, el campo al que se le agrego dicho índice y el tipo de índice.

7.- *Vistas y Funciones*: Las vistas y funciones se encuentran descritas en el diccionario de datos, en la sección de funciones y en la sección de vistas, donde podemos observar el nombre de la función o vista y la descripción de cada una.