

Trabajo Práctico de Gestión de Datos

Estrategia

Grupo LOS_GEDDES						
Cátedra	Ing. en Sistemas de Información	Nombre	Apellido	Legajo	Mail	Curso
 Integrantes del Grupo:		Jean Pierre	Chero Pomaleque	152.233-4	pierchero@gmail.com	K3671
		Matías Leandro	Anzorandía	168.063-8	matias.l.anzorandia@gmail.com	K3572
		Emiliano	González	135.246-5	gonzalemi42@gmail.com	K3572
		Paul Roussel	Vasquez	160.602-5		

Fecha de Presentación: 25/10/2020

Fecha Devolución: _____

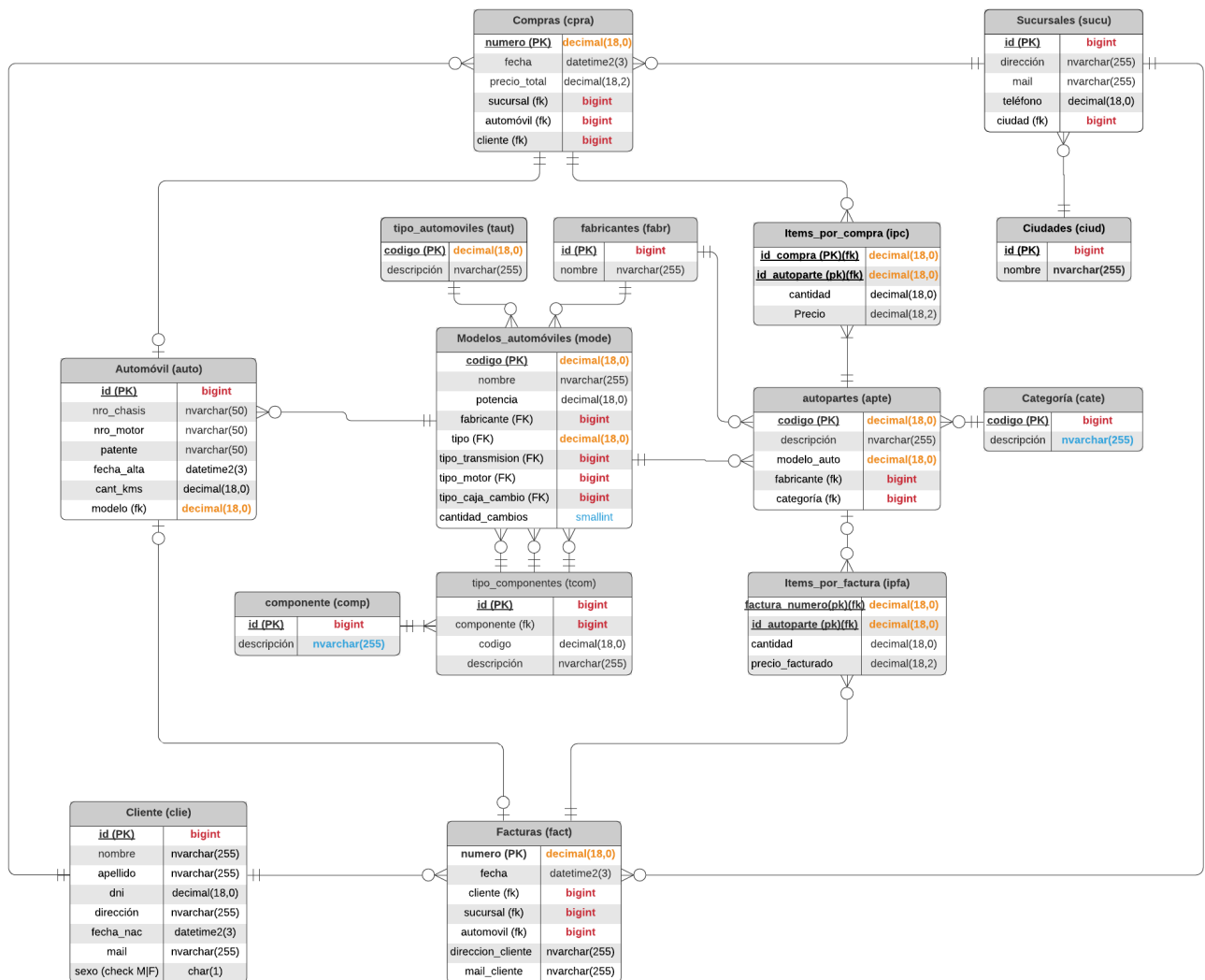
Calificación: _____

Firma Docente: _____

Índice

- Diagrama de Entidad-Relación Página 01
- Decisiones tomadas Página 02

Diagrama de Entidad-Relación



ESTRATEGIAS

Modelo De Datos

1. **Cliente – Compra:** Realizamos la relación cliente con compra.
2. **Tabla Paramétrica:** Creación de una tabla 'tipo_componentes' en la cual persisten todos los tipos de componentes asociados a automóvil en lugar de crear una tabla por cada componente. Este mismo se relaciona con la tabla componentes, el cual identifica que tipo de componente representa. (por ej: 'TRANSMISION', 'CAJA' y 'MOTOR'). Esto nos ahorra estar creando tablas por cada componente que se agregue al automóvil.
3. **Restricciones de Datos:** Declaramos la mayoría de campo con NOT NULL para poder asegurar la integridad y la completitud de los datos ajustándose al nuevo modelo del negocio.
4. **Tipado de Datos:** Declaramos tipo de dato BIGINT a los identificadores de las tablas para ampliar la longitud de la clave primaria
5. **Tablas ITEMS:** Creación de las tablas item_por_factura e item_por_compra para poder Normalizar los campos asociados a la facturación y compra. Tablas que usamos para el detalle de los mismos.

Performance y Optimización

1. **Union All:** Utilizamos union all para la unión de los select ante el insert de las tablas. La utilización del los union all, es la performance que tiene, ya que es mucho mas rápido no te realiza la comparación de los registros que te trae cada select, lo cual hace que sea mucho más rápido la unión.
2. **Constraints:** Declaramos las constraints al final de las creaciones de las tablas para poder agilizar el insert de los datos.
3. **JOINS:** Identificamos los campos únicos de las tablas para poder realizar la menor comparación en los JOINS. Esto nos evita estar matcheando muchos campos para poder identificar el mismo dato.
4. **Explain:** Utilización de plan de ejecución para poder medir y mejorar el costo de las queries.
5. **Claves Subrogadas:** Generación de claves subrogadas en tablas Paramétricas para mejorar el rendimiento de operaciones y poder

facilitar la lectura de datos mediante ese único identificador.

6. **Tablas Temporales:** Utilizamos tablas temporales para mejorar el rendimiento.
7. **Procedures:** Creación de Procedures, este nos da seguridad al ejecutar el insert de los datos, ya que al estar contenido dentro de un procedure nos asegura que nos va a ejecutar todo o nada, evitando así una tabla con inconsistencia. Por último, realizamos el dropeo de los mismos.