



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

ESTRATEGIA

LOS CRUD

Grupo 23

2024 - Primer Cuatrimestre

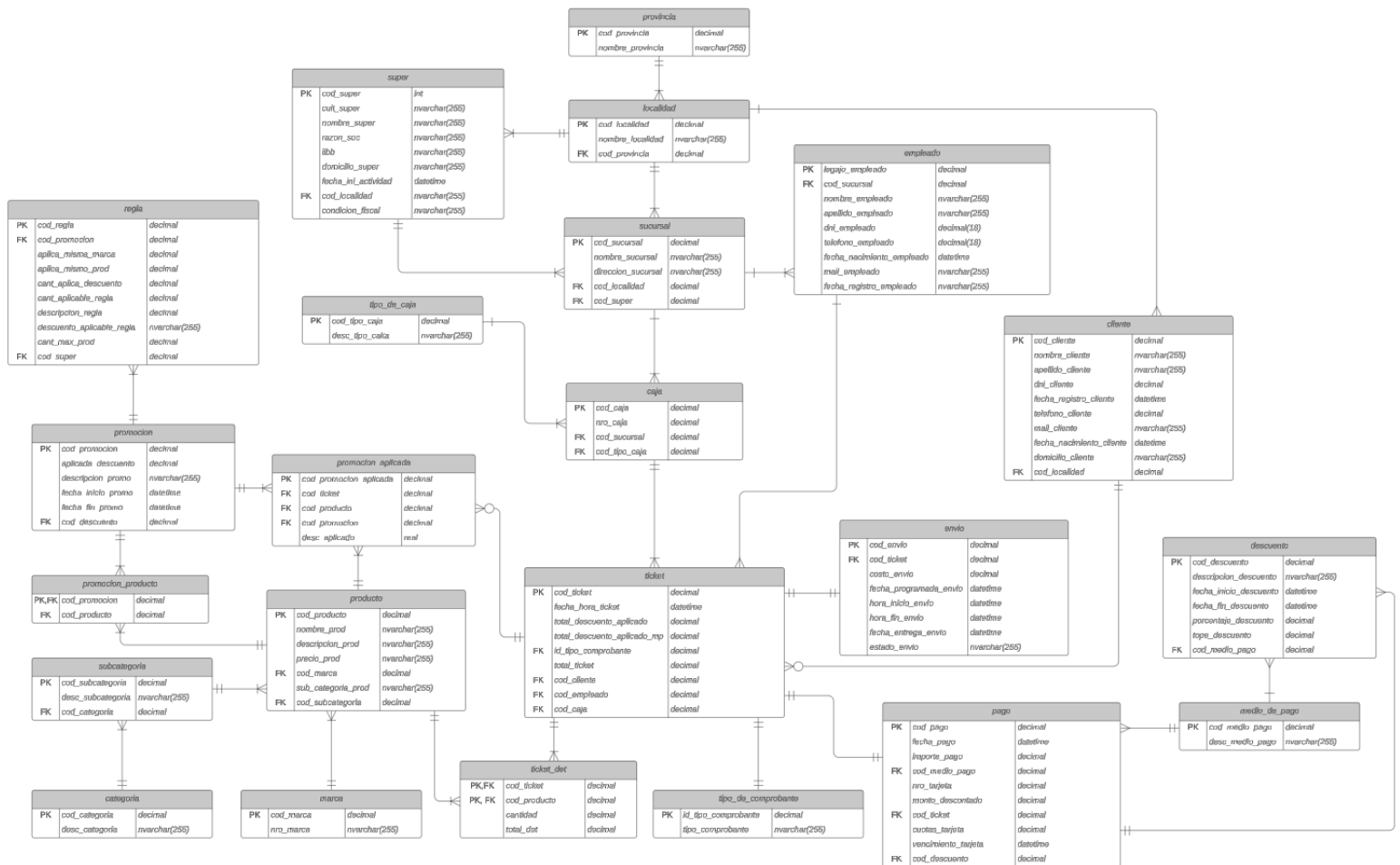
K3013

INTEGRANTE:	LEGAJO:
LORENZO, Melani	168.641-0
NEGRETE, Joaquin	204.599-0
LIZARRAGA GRIGONO, Tao	176.501-2
BERGES, Santiago	209.863-5



Diagrama Entidad-Relación

A partir de los requerimientos y casos de uso mencionados en el enunciado, se ha diseñado un DER con el fin de dar una nueva estructura relacional a los datos que se proporcionarán. A continuación, se explicará cómo se decidieron las principales tablas y relaciones (Foreign Keys) de las mismas.



Localidad:

Esta tabla hace uso de una foreign key:

cod_provincia: Cada provincia tiene varias localidades, es una relación de muchos a uno.

Ticket:

Esta tabla hace uso de cuatro foreign keys:



id_tipo_comprobante: Debido a que los tickets tienen uno y solo un tipo de comprobante, la relación 1-1 con *tipo_de_comprobante* será necesaria.

cod_cliente: Cada compra que un cliente haga generará un ticket, por lo que un cliente puede tener muchos tickets, por ende la relación es de muchos a uno.

cod_empleado: Los empleados son quienes generan los tickets, un ticket está generado por un empleado en particular por lo cual la relación es de muchos a uno.

nro_caja: Los tickets son emitidos en una caja, pero cada caja genera una cantidad de tickets, por lo cual en este caso también hay una relación de muchos a uno.

Pago:

Esta tabla hace uso de cuatro foreign keys:

cod_medio_pago: Existen distintos medios de pago. Al ejecutarse un pago se hace por un medio en particular, por lo tanto al usarse el mismo medio de pago para muchos pagos, la relación es de muchos a uno.

nro_tarjeta: En aquellos casos donde el pago sea electrónico, se suele requerir asociar una tarjeta.

cod_cliente: Este dato es obtenido por la asociación del ticket al cliente.

cod_ticket: Naturalmente existirá una relación 1 a 1 entre un ticket y su referido pago.

Promoción aplicada:

Esta tabla hace uso de tres foreign keys:

cod_ticket: Las promociones se aplican sobre los tickets, la misma promoción puede aplicar para varios por lo que la relación es muchos a uno.

cod_producto: Las promociones dependen del producto que aparezca en el ticket

cod_promocion: Las distintas promociones están en la tabla de promocion, por lo tanto esta relación es indispensable.

Ticket det:

Esta tabla hace uso de dos primary/foreign keys. Esta tabla intermedia existe porque sino habría una relación muchos a muchos entre ticket y producto, ya que un ticket puede tener muchos productos y los productos pueden estar presentes en muchos tickets. Dichas PK/FK son relaciones con dichas tablas.

Subcategoria:

Esta tabla hace uso de una foreign key:



cod_categoria: Al igual que en el caso de localidad/provincia, cada categoría puede tener varias subcategorías, resultando en esta relación entre tablas (muchos a uno).

Producto:

Esta tabla hace uso de dos foreign keys:

cod_marca: Cada producto está fabricado por una marca en concreto, cada marca pudiendo tener varios productos, por lo que hay una relación muchos a uno.

cod_subcategoria: Al igual que en los casos de localidad/provincia, muchas subcategorías pueden ser parte de una categoría y muchos productos pueden ser parte de una subcategoría.

Caja:

Esta tabla hace uso de dos foreign keys:

cod_sucursal: Cada sucursal tiene una cantidad determinada de cajas, por lo tanto es una relación de muchos a uno.

cod_tipo_caja: Los tipos de caja están tipificados, por lo cual se requiere esta relación (muchos a uno), donde un tipo puede estar asignado a muchas cajas.

Regla:

Esta tabla hace uso de dos foreign keys:

cod_promocion: Cada promoción está dada por cumplimiento de ciertas reglas, por lo que habrá una relación de muchos a uno.

cod_super: Las reglas varían según el supermercado, por lo cual esta relación es necesaria.

Sucursal:

Esta tabla hace uso de dos foreign keys:

cod_localidad: Así como cada provincia tiene muchas localidades, cada localidad puede tener varias sucursales, relación de muchos a uno.

cod_super: Cada supermercado puede tener varias sucursales, por lo que se da una relación de muchos a uno.

Super:

Esta tabla hace uso de una foreign key:

cod_localidad: Cada significado estará en una localidad en concreto, por lo que se requiere de esta relación.

Empleado:



Esta tabla hace uso de una foreign key:

cod_sucursal: Cada empleado trabaja en una sucursal en concreto, por lo cual, esta relación es indispensable.

Cliente:

Esta tabla hace uso de una foreign key:

cod_localidad: Entre los datos de los clientes se encontrará su domicilio, lo cual implica la relación con localidad, siendo de muchos a uno.

Envio:

Esta tabla hace uso de una foreign key:

cod_ticket: Para enviarse las compras, se requiere de los datos de los productos que se han comprado y los mismos se encuentran en el ticket asociado a la compra.

Descuento:

Esta tabla hace uso de una foreign key:

cod_medio_pago: Cada medio de pago puede tener una cierta cantidad de descuentos, por lo que se dará esta relación muchos a uno.

Promocion_producto:

Esta tabla hace uso de dos primary/foreign key, caso similar al de ticket_det. Al poder haber varias promociones para los mismos productos y varios productos en las mismas promociones, se genera la necesidad de esta tabla intermedia.

Promocion:

Esta tabla hace uso de una foreign key:

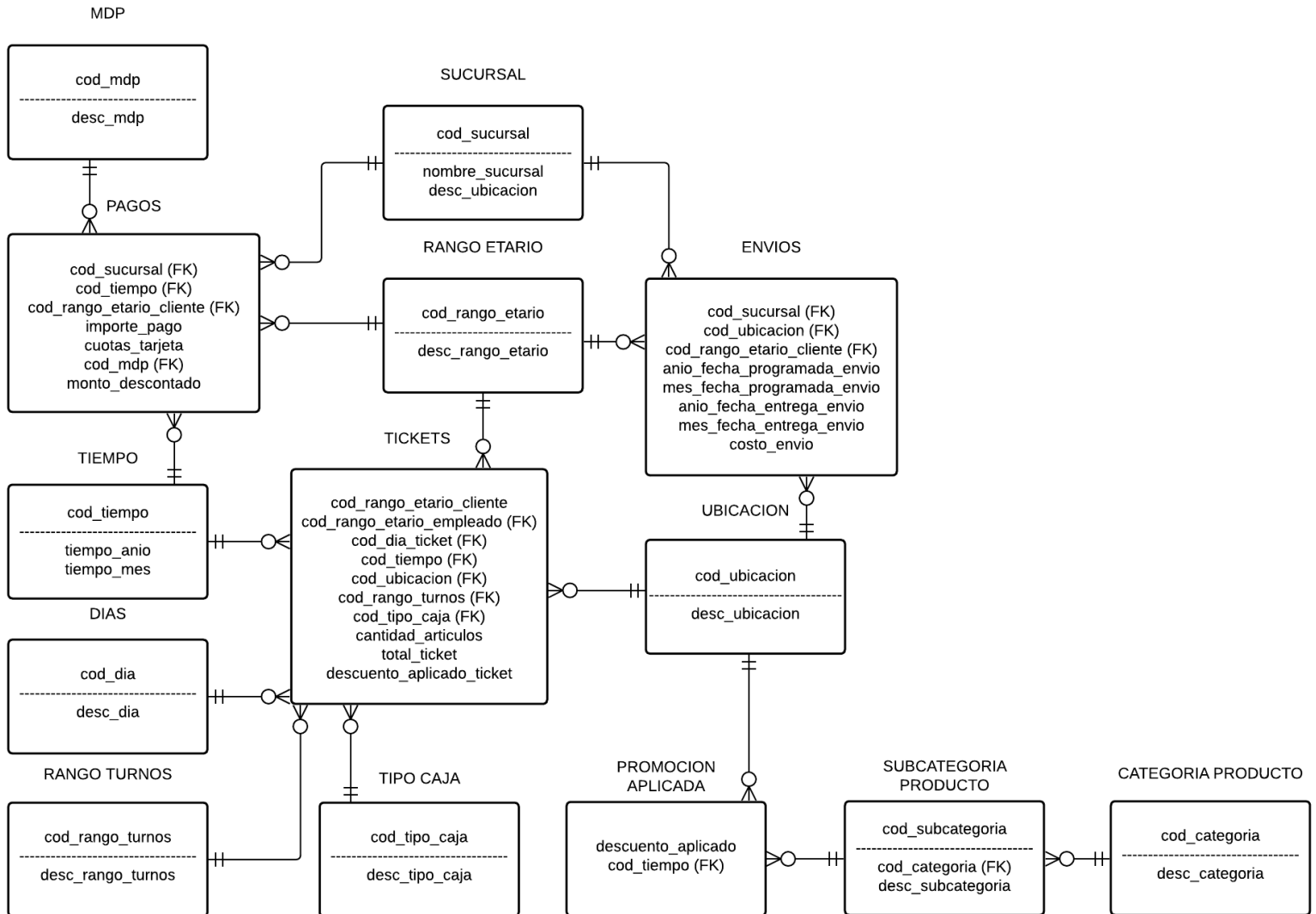
cod_descuento: Las promociones aplican un determinado descuento, por lo tanto existe la necesidad de esta relación.

Business Intelligence (BI)

Diagrama Entidad-Relación



Para Business Intelligence (BI a partir de ahora) se ha diseñado el siguiente Diagrama Entidad-Relación. El mismo será necesario para la realización de las vistas requeridas para este trabajo. A continuación, se explicará cómo se decidieron las tablas de hechos y relaciones (Foreign Keys) de las mismas.



Tickets:

Esta tabla hace uso de seis foreign keys:

cod_rango_etario_empleado: Esta foreign key será necesaria para implementar la vista 3.

cod_dia_ticket: Esta foreign key es un campo requerido para los Tickets.

cod_tiempo: Esta foreign key será necesaria para implementar las vistas 1, 2, 3, 4 y 5.

cod_ubicacion: Esta foreign key será necesaria para implementar las vistas 1 y 4.

cod_rango_turnos: Esta foreign key será necesaria para implementar las vistas 2 y 4.



cod_tipo_cajas: Esta foreign key será necesaria para implementar la vista 3.

Promoción Aplicada:

Esta tabla hace uso de una foreign key:

cod_tiempo: Esta foreign key será necesaria para implementar la vista 6.

Envios:

Esta tabla hace uso de tres foreign keys:

cod_sucursal: Esta foreign key será necesaria para implementar la vista 7.

cod_ubicacion: Esta foreign key será necesaria para implementar la vista 9.

cod_rango_etario_cliente: Esta foreign key será necesaria para implementar la vista 8.

Pagos:

Esta tabla hace uso de cuatro foreign keys:

cod_sucursal: Esta foreign key será necesaria para implementar la vista 10.

cod_tiempo: Esta foreign key será necesaria para implementar la vista 10 y 12.

cod_rango_etario_cliente: Esta foreign key será necesaria para implementar la vista 11.

cod_mdp: Esta foreign key será necesaria para implementar la vista 10 y 12.

Comentarios respecto implementación de vistas:

Vista 10: La vista entregada muestra las tres sucursales con mayor importe independientemente del medio de pagos porque es lo que interpretamos en un principio, pero aunque fuimos por esa opción, también surgió la interpretación de si lo que se debía mostrar eran las tres sucursales con mayor importe para *cada* medio de pago.