**TRABAJO FINAL INTEGRADOR**



**Estudiantes**

* PAGNUCCO MATEO
* BARANOVSKY FELIPE

**Materia**

* BASE DE DATOS 1

**Docente**

* GARCÍA MATTÍO MARIANO

**Fecha:**

* 10/06/2020

ÍNDICE

[**INTRODUCCIÓN**](#_fnqa23pfq1d7) **2**

[**DESARROLLO**](#_9r7p1pln2osz) **3**

[QUIENES SOMOS](#_zi3vazptsc6n) 3

[Historia](#_8e7et7t0rvco) 3

[CLIENTE](#_yo4ijg9mil7) 4

[PROBLEMÁTICA](#_i326ffa4uytn) 6

[Requerimientos:](#_d2e3g4z6wfdm) 6

[RESOLUCIÓN](#_p15x62ek5aq9) 9

[Modelo Conceptual:](#_ls24sz5bbtq7) 9

[Modelo Relacional:](#_3ia5l9iy7cml) 10

[Diagrama entidad-relación:](#_kuoad5bldrsp) 13

[Diccionario de datos:](#_8ktiiiybcd7y) 14

[Consultas sql:](#_gna7tms1rqyv) 28

# 

# 

# INTRODUCCIÓN

El presente es un trabajo elaborado por los estudiantes “Pagnucco Mateo” y “Felipe Baranovsky”, cuyo fin es textualizar y presentar a una empresa totalmente ficticia con su problemática y a la empresa formada por los estudiantes que se encargará de solucionarlo.

El trabajo se llevará a cabo de la siguiente manera:

1. Se presentará a la empresa elaborada por los estudiantes, nombre, misión, servicio que brinda, logo, etc.
2. Se presentará a la empresa que solicita los servicios de nuestra empresa, nombre, actividades, estructura, servicio que brinda.
3. Se explicará la problemática y lo que le solicita una empresa a otra
4. Se mostrará la solución a dicha problemática.

# DESARROLLO

## QUIENES SOMOS

Nedeah Analytia es una empresa informática que desarrolla soluciones enfocadas a las bases de datos relacionales SQL. La misión de nuestra empresa es la de ayudar a empresas a progresar mediante el uso de bases de datos relacionales.

En nuestra empresa creemos que somos capaces de afrontar cualquier desafío, por ello nuestro eslogan es:

*“Diseñar las bases de su base de datos no es un dato nuevo para nosotros”*

### Historia

Nedeah Analytia tiene sus orígenes el 20 de octubre del 2000. Fundada por los hermanos Tony; el empresario Tony Stark y el agente de negocios Tony Matar.

La empresa presentó un duro comienzo debido a la crisis internacional que se afrontaba por aquella fecha, pero a poco a poco, a medida que el mundo se recuperaba, Nedeah Analytia creció desenfrenadamente y se puso en el punto de mira de todas aquellas empresas que adquirían estabilidad económica y buscaban actualizar su sistema. Aquellas empresas que administraban sus datos desde papel o documentos excel, querían dar un salto e invertir en el armado de una base de datos.

Empresas como Valve, Arcor, Guaymallén, Don Hugo S.A., From Software, entre otras, decidieron apostar por Nedeah Analytia y confiaron la confección de sus bases de datos a esta empresa.

## CLIENTE

H.C.S.A. S.A.



**H:** Health

**C:** Compromise

**S:** Security

**A:** Attention

*H.C.S.A.* (Health, Compromise, Security, and Attention) es la empresa que contrata a *Nedeah Analytia*. Esta empresa se dedica a la fabricación de respiradores y barbijos. A su vez, tiene una única planta ubicada en la provincia de Córdoba en donde se fabrican tanto barbijos como respiradores. Con la reciente tragedia acontecida a nivel mundial, las ventas de esta empresa incrementaron con creces. Esta pequeña empresa, pasó a ser una gran empresa de repente y en estos momentos necesita adaptarse a la realidad.

La empresa no solo está intentando aumentar las plantas de producción y la cantidad de empleados y máquinas, sino que busca pasar su organización de sistema en excel a una base de datos relacional.

Para esto, la empresa le brindó la siguiente información de su estructura a *Nedeah Analytia:*



## PROBLEMÁTICA

La reconocida empresa H.C.S.A. S.A. necesita aumentar su capacidad de producción debido a su crecimiento reciente por la pandemia a nivel mundial. Para ello, precisa de una mayor organización y una mejor disposición de sus datos. Hasta la fecha, la empresa ha estado almacenando sus datos en hojas de cálculo Excel, pero para permitir su crecimiento deben migrar su sistema a una base de datos.

Por este motivo, H.C.S.A. S.A. ha contratado los servicios de *Nedeah Analytia*, confiando en esta famosísima compañía e indiscutible líder en el ámbito de la elaboración de bases de datos.

### Requerimientos:

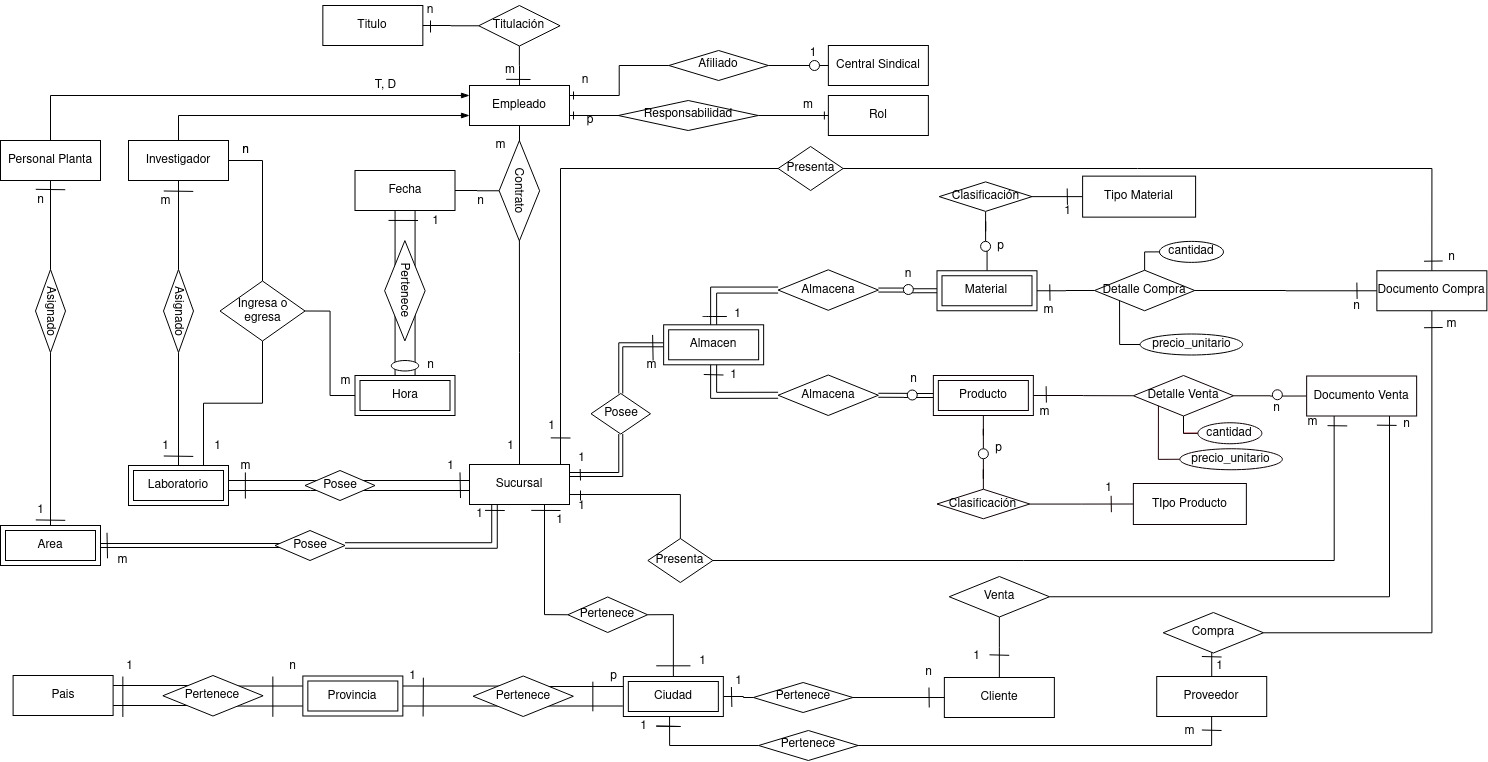
La empresa ha sido muy clara respecto a lo que busca que contenga su base de datos y para esto, ha compartido información de sus funciones y estructura:

1. Se desea saber donde está ubicada la sucursal, el nombre, su código de sucursal, domicilio y número de teléfono. (Se pide esto a pesar de tener una única sucursal en Córdoba ya que se tiene muy en cuenta futuras sucursales). Solo se permitirá una sucursal por ciudad.
2. Se desea conocer información de las distintas áreas de trabajo, de que area se trata, cantidad de gente que trabaja en dicha área, número de teléfono y el código de área. Recordar que se puede tratar tanto del área de administración, técnica, industrial, de ventas y recursos humanos.
3. Las sucursales cuentan con dos laboratorios donde trabajan los investigadores. Estos no pueden trabajar en más de un laboratorio. Además se quiere tener un control de la hora y fecha en la que un empleado entra o sale del laboratorio. También, de qué uso se le da a ese laboratorio, para investigación de barbijos o para investigación de respiradores.
4. En esta empresa, existen distintos tipos de empleados. Estos se dividen en investigadores (aquellos que realizan sus actividades en el laboratorio) y personal de planta (ya sean del área de administración, técnica, industrial, de ventas o recursos humanos), cabe destacar que un empleado solo puede ser de un tipo de empleado en un momento dado.
5. Se busca tener un control histórico de las distintas contrataciones de los empleados en las diferentes sucursales. Considerando que un empleado puede ser contratado múltiples veces.
6. Cada personal de planta puede tener múltiples roles asignados (teniendo al menos un rol). El rol puede tratarse ya sea de administrador, contador, ingeniero, etc. El Rol también guarda información sobre el sueldo base. Por su parte, los investigadores tienen un único rol, y un sueldo fijo.
7. De todos los empleados se desea conocer el DNI, nombre, apellido, numero de telefono y fecha en la que fue contratado. También se quiere conocer los títulos académicos de los mismos (se debe tener al menos uno). Cabe destacar, que algunos empleados están afiliados a alguna central sindical. Se quiere saber el nombre de la central sindical a la cual están afiliados y su número de afiliado.
8. Por otra parte, se desea conocer la información de los proveedores, tanto su nombre como código de proveedor y ciudad a la que pertenece. Además se desea conocer los mismos datos de los clientes de la empresa.
9. Cada sucursal tiene múltiples almacenes, y además se desea saber la cantidad de materiales y de productos disponibles. De los materiales se quiere conocer su nombre e id. En tanto a los productos, se desea conocer su nombre, id y precio actual.
10. Se desea conocer la información de las facturas tanto de compra como de venta. Las facturas de compra deberán contener información de la cantidad, el código del material, el código del proveedor y el precio por unidad. Por otro lado, la factura de venta tiene información de la cantidad, código de cliente, código del producto y el precio al que se lo vendió. Cabe destacar que ambas facturas contienen un código de factura, numero de factura, un código de sucursal, fecha y si se encuentran anuladas o no.

## RESOLUCIÓN

Según los requerimientos planteados por la empresa “H.C.S.A.”, se llegaron a los siguientes modelos:

### Diagrama del Modelo Conceptual:



Luego se trasladó de del modelo conceptual al modelo relacional, resultando de la siguiente manera:

### Modelo Relacional:

PAÍS (id-país, país)

PROVINCIA (id-provincia, id-país, provincia)

donde {id-país} referencia PAÍS.

CIUDAD (id-pais, id-provincia, id-ciudad, ciudad)

donde {id-país} referencia PAÍS,

donde {id-provincia} referencia PROVINCIA.

SUCURSAL (id-sucursal, id-pais, id-provincia, id-ciudad, sucursal, dirección, teléfono)

donde {id-ciudad, id-provincia, id-pais} referencia CIUDAD.

FECHA (fecha)

HORA (fecha, hora)

donde {fecha} referencia FECHA.

EMPLEADO (id-empleado, nombre, apellido, DNI, teléfono, id-central-sindical)

donde {id-central-sindical} referencia CENTRAL-SINDICAL

y id-central-sindical admite valores nulos.

CONTRATO (id-empleado, id-sucursal, fecha)

donde {id-empleado} referencia EMPLEADO,

donde {id-sucursal} referencia SUCURSAL,

donde {fecha} referencia FECHA.

TÍTULO (id-título, titulo)

TITULACIÓN (id-título, id-empleado)

donde {id-título} referencia TITULO,

donde {id-empleado} referencia EMPLEADO.

ROL (id-rol, rol, sueldo)

RESPONSABILIDAD (id-rol, id-empleado)

dónde {rol,sueldo} referencia ROL,

donde {id-empleado} referencia EMPLEADO.

CENTRAL-SINDICAL (id-central-sindical, central-sindical)

ÁREA (id-sucursal, id-area, area, teléfono-interno)

donde {id-sucursal} referencia SUCURSAL.

LABORATORIO (id-sucursal, id-laboratorio, uso)

donde {id-sucursal} referencia SUCURSAL.

PERSONAL-PLANTA (id-empleado, id-area)

donde {id-empleado} referencia EMPLEADO,

donde {id-area} referencia ÁREA.

INVESTIGADOR (id-empleado, id-laboratorio)

donde {id-empleado} referencia EMPLEADO,

donde {id-laboratorio} referencia LABORATORIO.

INGRESO-EGRESO (id-empleado, fecha, hora, id-laboratorio, id-sucursal, ingreso)

donde {id-empleado} referencia EMPLEADO,

donde {id-laboratorio, id-sucursal} referencia LABORATORIO,

donde {fecha, hora} referencia HORA.

PROVEEDOR (id-proveedor, proveedor, teléfono, id-ciudad, id-provincia, id-país)

donde {id-ciudad, id-provincia, id-país} referencia CIUDAD.

CLIENTE (id-cliente, cliente, telefono, id-ciudad, id-provincia, id-país)

donde {id-ciudad, id-provincia, id-país} referencia CIUDAD.

ALMACÉN (id-sucursal, id-almacen)

donde {id-sucursal} referencia SUCURSAL.

MATERIAL (id-almacen, id-sucursal, id-material, id-tipo-material, material, cantidad)

donde {id-almacen, id-sucursal} referencia ALMACEN,

donde {id-tipo-material} referencia TIPO-MATERIAL.

PRODUCTO (id-almacen, id-sucursal, id-producto, id-tipo-producto, producto, precio-unitario, cantidad)

donde {id-almacen, id-sucursal} referencia ALMACEN,

donde {id-tipo-producto} referencia TIPO-PRODUCTO.

TIPO-MATERIAL (id-tipo-material, tipo)

TIPO-PRODUCTO (id-tipo-producto, tipo)

DETALLE-COMPRA (id-factura-compra, id-material, id-almacen, id-sucursal cantidad, precio-unitario)

donde {id-factura-compra} referencia DOCUMENTO-COMPRA,

donde {id-material, id-almacen, id-sucursal} referencia MATERIAL.

DETALLE-VENTA (id-factura-venta, id-producto, id-almacen, id-sucursal, cantidad, precio-unitario)

donde {id-factura-venta} referencia DOCUMENTO-VENTA.

donde {id-producto, id-almacen, id-sucursal} referencia PRODUCTO.

DOCUMENTO-COMPRA (id-factura-compra, id-proveedor, id-sucursal, fecha, anulada)

donde {id-proveedor} referencia a PROVEEDOR,

donde {id-sucursal} referencia SUCURSAL.

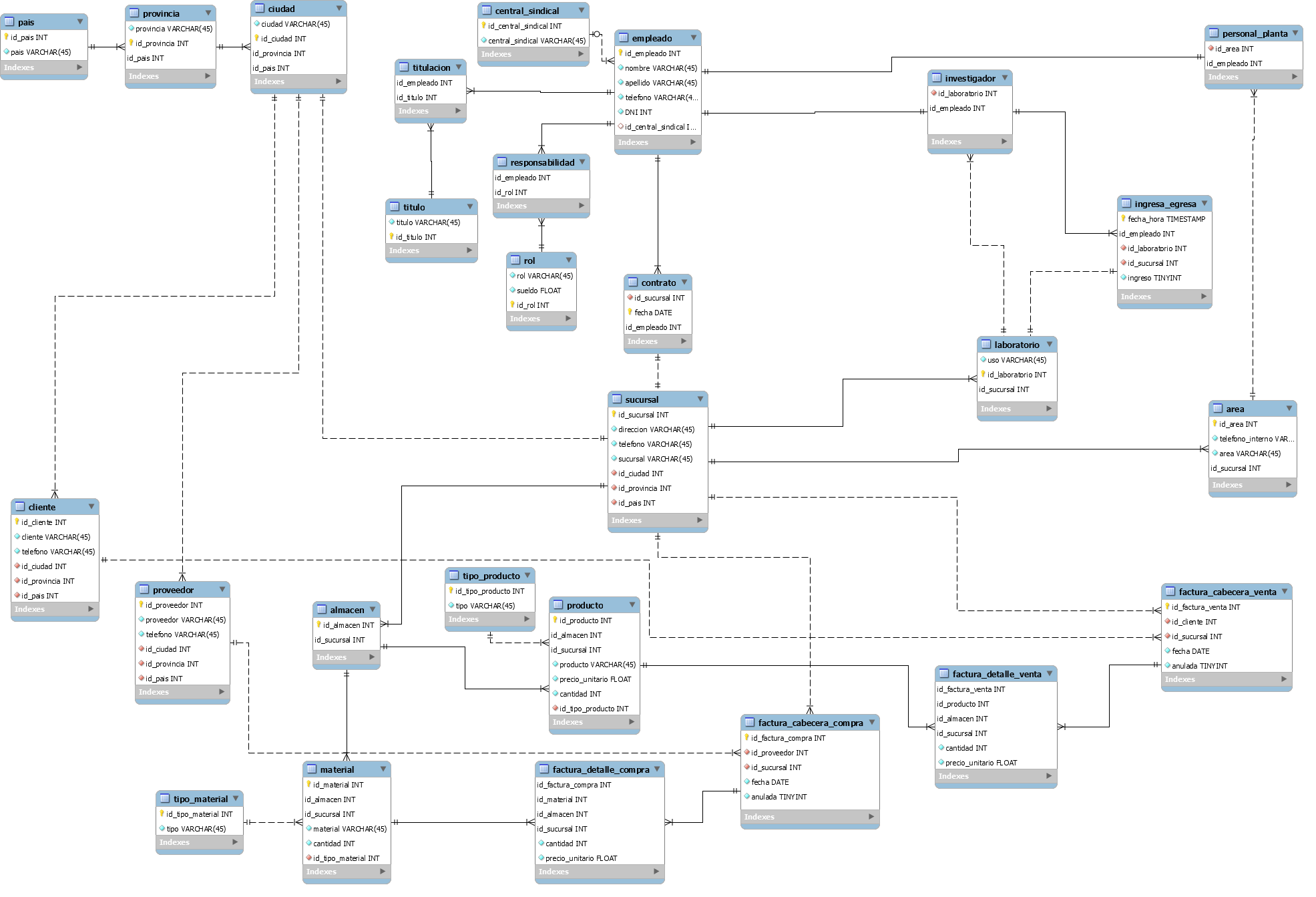
DOCUMENTO-VENTA (id-factura-venta, id-cliente, id-sucursal, fecha, anulada)

donde {id-cliente} referencia a PROVEEDOR,

donde {id-sucursal} referencia SUCURSAL.

Con ayuda de la herramienta de trabajo MySQL Workbench, basándose en los anteriores, se desarrolló el siguiente diagrama entidad-relación:

### Diagrama entidad-relación:

**

Para un mejor entendimiento de este se realizó el siguiente diccionario de datos:

### Diccionario de datos:

| Tabla **pais**: almacena información de los países que tienen al menos una sucursal. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-pais | int |  | x |  |  | x | x | Permite identificar de manera más eficaz el país. |
| pais | varchar | 45 |  | x |  | x |  | Nombre del país. |

| Tabla **provincia**: almacena la información de las provincias que tienen al menos una sucursal. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-provincia | int |  | x |  |  | x | x | Permite identificar de manera más eficaz las provincias. |
| id-pais | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: pais | x |  |  |
| provincia | varchar | 45 |  |  |  | x |  | Nombre de la provincia. |

| Tabla **ciudad**: almacena la información de las ciudades que tienen una sucursal. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-ciudad | int |  | x |  |  | x | x | Permite identificar de manera más eficaz las ciudades. |
| id-pais | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: provincia | x |  |  |
| id-provincia | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: provincia | x |  |  |
| ciudad | varchar | 45 |  |  |  | x |  | Nombre de la ciudad. |

| Tabla **sucursal**: almacena la información de las distintas sucursales. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-sucursal | int |  | x |  |  | x | x | Permite identificar de manera sencilla las sucursales. |
| id-pais | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  |  |
| id-provincia | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  |  |
| id-ciudad | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  | Es posible que existan en un futuro sucursales en distintas ciudades del mundo. |
| direccion | varchar | 45 |  |  |  | x |  |  |
| telefono | varchar | 45 |  | x |  | x |  |  |
| sucursal | varchar | 45 |  |  |  | x |  | Nombre de la sucursal. |

| Tabla **central\_sindical**: almacena la información de las centrales sindicales a las cuales están afiliados por lo menos un empleado. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-central-sindical | int |  | x |  |  | x | x | Permite identificar de manera más eficaz a las central sindical. |
| central-sindical | varchar | 45 |  |  |  | x |  |  |

| Tabla **empleado**: almacena la información de los títulos académicos que posee por lo menos un empleado. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-empleado | int |  | x |  |  | x | x | Permite identificar de manera más sencilla a los distintos empleados. |
| id-central-sindical | int |  |  |  | DELETE: set null,UPDATE: set null, Tabla madre: central\_sindical |  |  | Un empleado puede o no estar afiliado a una central sindical. |
| nombre | varchar | 45 |  |  |  | x |  |  |
| apellido | varchar | 45 |  |  |  | x |  |  |
| telefono | varchar | 45 |  |  |  | x |  |  |
| DNI | int |  |  | x |  | x |  |  |

| Tabla **contrato**: almacena la información de las contrataciones de los empleados por las sucursales a través del tiempo. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| fecha | date |  | x |  |  | x |  | Fecha en la cual se realiza la contratación. |
| id-empleado | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: empleado | x |  | Se guarda la información del empleado contratado. |
| id-sucursal | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: sucursal | x |  | Se guarda la información de la sucursal a la cual fue contratado un empleado. |

| Tabla **rol**: almacena la información de los roles que posibles para un empleado. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-rol | int |  | x |  |  | x | x | Permite identificar de manera más eficaz a las central sindical. |
| sueldo | float |  |  |  |  | x |  | Sueldo del rol en pesos argentinos. |
| rol | varchar | 45 |  |  |  | x |  |  |

| Tabla **responsabilidad**: almacena la información de los títulos académicos por cada empleado. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-empleado | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: empleado | x |  |  |
| id-rol | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: rol | x |  |  |

| Tabla **titulo**: almacena la información de los títulos académicos que posee por lo menos un empleado. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-titulo | int |  | x |  |  | x | x | Permite identificar de manera más eficaz los títulos académicos. |
| titulo | varchar | 45 |  |  |  | x |  |  |

| Tabla **titulacion**: almacena la información de los títulos académicos por cada empleado. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-empleado | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: empleado | x |  |  |
| id-titulo | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: titulo | x |  |  |

| Tabla **laboratorio**: almacena la información de los laboratorios de una sucursal. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-laboratorio | int |  | x |  |  | x | x | Permite identificar de manera más eficaz los laboratorios. |
| id-sucursal | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: sucursal | x |  | Los laboratorios se separan por sucursal |
| uso | varchar | 45 |  |  |  | x |  | Se refiere a la finalidad del laboratorio |

| Tabla **area**: almacena la información de las distintas áreas de una sucursal. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-area | int |  | x |  |  | x | x | Permite identificar de manera más sencilla a las distintas áreas. |
| id-sucursal | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: sucursal | x |  | Las áreas se separan por sucursal. |
| telefono-interno | varchar | 45 |  | x |  | x |  |  |
| area | varchar | 45 |  |  |  | x |  |  |

| Tabla **investigador**: almacena la información de los investigadores. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-empleado | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: empleado | x |  | Permite identificar de manera más eficaz los investigadores. |
| id-laboratorio | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: laboratorio | x |  | Los investigadores trabajan en un laboratorio. |

| Tabla **personal\_planta**: almacena la información del personal de planta. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-empleado | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: empleado | x |  | Permite identificar de manera más eficaz al personal de planta. |
| id-area | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: area | x |  | El personal de planta esta separado por area en donde trabaja. |

| Tabla **ingresa\_egresa**: almacena la información del registro de la fecha y hora de la entrada y salida de los investigadores en los laboratorios. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| fecha-hora | timestamp |  | x |  |  | x |  | Indica la fecha y hora en la que un investigador entra o sale de un laboratorio. |
| id-empleado | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: investigador | x |  | Se guarda la información del investigador. |
| id-laboratorio | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: laboratorio | x |  | Se guarda información del laboratorio al cual el investigador ingresa o egresa. |
| id-sucursal | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: laboratorio | x |  | Se guarda información de la sucursal a la cual pertenece dicho laboratorio. |
| ingreso | tinyint |  |  |  |  | x |  | Si su valor es uno, indica que el investigador entra a un laboratorio, si es cero indica que el investigador sale. |

| Tabla **almacen**: almacena la información de los almacenes por sucursal. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-almacen | int |  | x |  |  | x | x | Permite identificar rápidamente a los almacenes. |
| id-sucursal | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: sucursal | x |  | Los almacenes se separan por sucursal |

| Tabla **cliente**: almacena la información de los clientes de la empresa. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-cliente | int |  | x |  |  | x | x | Permite identificar de manera más sencilla a los clientes. |
| cliente | varchar | 45 |  |  |  | x |  |  |
| telefono | varchar | 45 |  | x |  | x |  |  |
| id-ciudad | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  | Los clientes se separan por ciudad |
| id-provincia | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  |  |
| id-pais | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  |  |

| Tabla **proveedor**: almacena la información de los proveedores de la empresa. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-proveedor | int |  | x |  |  | x | x | Permite identificar de manera más eficaz a los proveedores. |
| proveedor | varchar | 45 |  |  |  | x |  |  |
| telefono | varchar | 45 |  | x |  | x |  |  |
| id-ciudad | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  | Los proveedores se separan por ciudad |
| id-provincia | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  |  |
| id-pais | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: ciudad | x |  |  |

| Tabla **tipo\_producto**: almacena la información de los posibles tipos de productos. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-tipo-producto | int |  | x |  |  | x | x | Permite ubicar los distintos tipos de productos de manera más eficaz. |
| tipo | varchar | 45 |  |  |  | x |  |  |

| Tabla **tipo\_material**: almacena la información de los posibles tipos de material. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-tipo-material | int |  | x |  |  | x | x | Permite ubicar los distintos tipos de material de manera más eficaz. |
| tipo | varchar | 45 |  |  |  | x |  |  |

| Tabla **material**: almacena la información de los materiales disponibles. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Presision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-material | int |  | x |  |  | x | x | Permite identificar a los distintos materiales. |
| id-almacen | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: almacen | x |  | Los materiales se separan por almacen. |
| id-sucursal | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: almacen | x |  | Los materiales se separan por sucursal. |
| id-tipo-material | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: tipo\_material | x |  | Los materiales se distinguen por tipo. |
| cantidad | int |  |  |  |  | x |  |  |
| material | varchar | 45 |  |  |  | x |  |  |

| Tabla **producto**: almacena la información de los productos disponibles. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-producto | int |  | x |  |  | x | x | Permite identificar a los distintos productos. |
| id-almacen | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: almacen | x |  | Los productos se separan por almacen. |
| id-sucursal | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: almacen | x |  | Los productos se separan por sucursal. |
| id-tipo-producto | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: tipo\_producto | x |  | Los productos se distinguen por tipo. |
| producto | varchar | 45 |  |  |  | x |  |  |
| cantidad | int |  |  |  |  | x |  |  |
| precio-unitario | float |  |  |  |  | x |  |  |

| Tabla **factura\_cabecera\_venta**: almacena la información de la cabecera de las facturas relacionada con la venta de productos. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-factura-venta | int |  | x |  |  | x | x | Permite identificar de manera más eficaz a las facturas de ventas. |
| id-cliente | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: cliente | x |  | Información del cliente al cual se le está vendiendo. |
| id-sucursal | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: sucursal | x |  | Las facturas se separan por sucursales. |
| fecha | date |  |  |  |  | x |  | Fecha de facturación de la misma. |
| anulada | tinyint |  |  |  |  | x |  | Si su valor es uno, indica que la factura está anulada, si es cero indica que la factura no está anulada. |

| Tabla **factura\_cabecera\_compra**: almacena la información de la cabecera de las facturas relacionada con la compra de productos. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-factura-compra | int |  | x |  |  | x | x | Permite identificar de manera más eficaz a las facturas de compras. |
| id-proveedor | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: proveedor | x |  | Información del proveedor al cual se le está comprando. |
| id-sucursal | int |  |  |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: sucursal | x |  | Las facturas se separan por sucursales. |
| fecha | date |  |  |  |  | x |  | Fecha de facturación de la misma. |
| anulada | tinyint |  |  |  |  | x |  | Si su valor es uno, indica que la factura está anulada, si es cero indica que la factura no está anulada. |

| Tabla **factura\_detalle\_venta**: almacena la información del detalle de las facturas relacionada con la venta de productos. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-factura-venta | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: factura\_cabecera\_venta | x |  | Código de la factura cabecera correspondiente. |
| id-producto | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: producto | x |  |  |
| id-almacen | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: producto | x |  | Las facturas se separan por almacen. |
| id-sucursal | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: producto | x |  | Las facturas se separan por sucursales. |
| cantidad | int |  |  |  |  | x |  | Cantidad de productos vendidos en una factura. |
| precio-unitario | float |  |  |  |  | x |  | Precio por producto. |

| Tabla **factura\_detalle\_compra**: almacena la información del detalle de las facturas relacionada con la compra de materiales. | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre | Tipo | Precision / Formato | PK | UQ | FK | Not Null | AI | Observaciones y consideraciones importantes |
| id-factura-compra | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: factura\_cabecera\_compra | x |  | Código de la factura cabecera correspondiente. |
| id-material | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: material | x |  |  |
| id-almacen | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: material | x |  | Las facturas se separan por almacen. |
| id-sucursal | int |  | x |  | DELETE: Cascade,UPDATE: Cascade, Tabla madre: material | x |  | Las facturas se separan por sucursales. |
| cantidad | int |  |  |  |  | x |  | Cantidad de productos vendidos en una factura. |
| precio-unitario | float |  |  |  |  | x |  | Precio por producto. |

A continuación se realizaron diez consultas de ejemplo que podrían ser usadas por la empresa para gestionar la base de datos ya planteada.

### Consultas sql:

* La empresa busca conocer su situación en cuanto a ventas. Para esto, hace un análisis de mercado y desea conocer alguna forma de obtener mayores ganancias. Se desea conocer la información del importe vendido y la cantidad vendida del producto más y menos vendido, por tipo de producto. [Tipo, Producto, Recaudado, Cantidad\_Vendida]

De esta forma se puede saber cual es el producto que deja mayores ganancias y cual es el que deja menor ganancia, teniendo como objetivo focalizarse en el producto que deja mayores ganancias y centrarse en la producción de este.

Además, ver cual es el producto que deja menor ganancias y analizar la situación y ver alguna forma de incrementar su rentabilidad o en su defecto retirarlo del mercado.

| SELECT  ll.\* FROM  (SELECT  tp.tipo,  prod.producto,  SUM(fdv.cantidad \* fdv.precio\_unitario) Recaudado,  SUM(fdv.cantidad) Cantidad\_Vendido  FROM  tipo\_producto tp  INNER JOIN  producto prod  ON prod.id\_tipo\_producto = tp.id\_tipo\_producto  INNER JOIN  factura\_detalle\_venta fdv  ON fdv.id\_producto = prod.id\_producto  INNER JOIN  factura\_cabecera\_venta fcv  ON fcv.id\_factura\_venta = fdv.id\_factura\_venta  WHERE  fcv.anulada = 0  GROUP BY prod.producto, tp.tipo) ll  INNER JOIN  (SELECT  Tipo,  MAX(Recaudado) max,  MIN(Recaudado) min  FROM  (SELECT  tp.tipo Tipo,  prod.producto,  SUM(fdv.cantidad \* fdv.precio\_unitario) Recaudado,  SUM(fdv.cantidad) Cantidad\_Vendido  FROM  tipo\_producto tp  INNER JOIN  producto prod  ON prod.id\_tipo\_producto = tp.id\_tipo\_producto  INNER JOIN  factura\_detalle\_venta fdv  ON fdv.id\_producto = prod.id\_producto  INNER JOIN  factura\_cabecera\_venta fcv  ON fcv.id\_factura\_venta = fdv.id\_factura\_venta  WHERE  fcv.anulada = 0  GROUP BY prod.producto, tp.tipo) listProd  GROUP BY Tipo) maxMinTipo  ON maxMinTipo.tipo = ll.tipo AND (ll.Recaudado = maxMinTipo.max OR ll.Recaudado = maxMinTipo.min); |
| --- |

Por otra parte también ofrecemos otra versión en la cual se muestre la cantidad vendida y el importe por tipo de producto y además muestre cual es el producto con mayor ganancias vendido y cual es el de menos ganancias. De una forma tal que no se tenga que ni comparar los valores de ventas, sino simplemente ver en qué columna se encuentra, por otra parte, brindar información general de los tipos de productos y su situación en ventas.

[Tipo, CantidadVendida, Importe, ProductoMasVendido, ProductoMenosVendido]

| SELECT  tp.tipo Tipo,  SUM(fdv.cantidad) CantidadVendida,  SUM(fdv.cantidad\*fdv.precio\_unitario) Importe,  pm.producto ProductoMasVendido,  pl.producto ProductoMenosVendido FROM  factura\_detalle\_venta fdv  INNER JOIN  producto p  ON fdv.id\_producto = p.id\_producto  INNER JOIN  tipo\_producto tp  ON p.id\_tipo\_producto = tp.id\_tipo\_producto  INNER JOIN  factura\_cabecera\_venta fcv  ON fdv.id\_factura\_venta = fcv.id\_factura\_venta  LEFT JOIN  (SELECT  p.producto Producto,  tp.id\_tipo\_producto idTipo,  SUM(fdv.cantidad\*fdv.precio\_unitario) Importe  FROM  factura\_detalle\_venta fdv  INNER JOIN  factura\_cabecera\_venta fcv  ON fdv.id\_factura\_venta = fcv.id\_factura\_venta  INNER JOIN  producto p  ON fdv.id\_producto = p.id\_producto  INNER JOIN  tipo\_producto tp  ON p.id\_tipo\_producto = tp.id\_tipo\_producto  WHERE  fcv.anulada = 0  GROUP BY Producto, idTipo  ORDER BY Importe DESC LIMIT 1) pm  ON tp.id\_tipo\_producto = pm.idTipo  LEFT JOIN  (SELECT  p.producto Producto,  tp.id\_tipo\_producto idTipo,  SUM(fdv.cantidad\*fdv.precio\_unitario) Importe  FROM  factura\_detalle\_venta fdv  INNER JOIN  factura\_cabecera\_venta fcv  ON fdv.id\_factura\_venta = fcv.id\_factura\_venta  INNER JOIN  producto p  ON fdv.id\_producto = p.id\_producto  INNER JOIN  tipo\_producto tp  ON p.id\_tipo\_producto = tp.id\_tipo\_producto  WHERE  fcv.anulada = 0  GROUP BY Producto, idTipo  ORDER BY Importe ASC  LIMIT 1) pl  ON tp.id\_tipo\_producto = pl.idTipo GROUP BY tp.tipo, ProductoMasVendido, ProductoMenosVendido ORDER BY tp.tipo; |
| --- |

* El contador de la empresa Alberto Hernandez desea corroborar los importes de deber y haber de la empresa para el arqueo de caja. Para esto desea ver el total vendido y el total comprado por fecha. Ordenado por fecha.

[fecha, total vendido, total comprado]

| SELECT  Fechas.fecha,  TRUNCATE(SUM(fdv.cantidad \* fdv.precio\_unitario), 2) TotalVendido,  TRUNCATE(SUM(fdc.cantidad \* fdc.precio\_unitario), 2) TotalComprado  FROM  (SELECT  fcc.fecha   FROM  factura\_cabecera\_compra fcc   UNION DISTINCT   SELECT  fcv.fecha   FROM  factura\_cabecera\_venta fcv)AS Fechas   LEFT JOIN  (SELECT  \*   FROM  factura\_cabecera\_venta   WHERE  anulada = 0)fcv ON Fechas.fecha = fcv.fecha   LEFT JOIN  (SELECT  \*   FROM  factura\_cabecera\_compra   WHERE  anulada = 0) fcc ON Fechas.fecha = fcc.fecha   LEFT JOIN  factura\_detalle\_venta fdv ON fcv.id\_factura\_venta = fdv.id\_factura\_venta   LEFT JOIN  factura\_detalle\_compra fdc ON fcc.id\_factura\_compra = fdc.id\_factura\_compra  WHERE  fcc.anulada = 0 OR fcv.anulada = 0  GROUP BY Fechas.fecha  ORDER BY Fechas.fecha; |
| --- |

* La empresa está buscando zonas para instalar una nueva sucursal. Para esto desea conocer la información de cuánto se vendió por ciudad para conocer cuál es la que da más beneficios, ordenado por el importe vendido de mayor a menor. Para de esta manera instalar la sucursal en un lugar que asegure la máxima rentabilidad. [país, provincia, ciudad, vendido]

| SELECT   p.pais,  pr.provincia,  c.ciudad,  SUM(fdv.cantidad \* fdv.precio\_unitario) Vendido FROM  factura\_cabecera\_venta fcv  INNER JOIN  factura\_detalle\_venta fdv ON fcv.id\_factura\_venta = fdv.id\_factura\_venta  INNER JOIN  cliente cl ON fcv.id\_cliente = cl.id\_cliente  INNER JOIN  ciudad c ON cl.id\_ciudad = c.id\_ciudad  INNER JOIN  provincia pr ON c.id\_provincia = pr.id\_provincia  INNER JOIN  pais p ON pr.id\_pais = p.id\_pais WHERE  fcv.anulada = 0 GROUP BY c.ciudad , pr.provincia , p.pais ORDER BY Vendido DESC; |
| --- |

* La empresa está buscando contratar a más empleados, para esto quiere ver en donde es primordial una mayor cantidad de personal. Desea saber los datos de todos los empleados de la sucursal. Si el empleado es un investigador se desea saber el laboratorio al que está asignado, y en el caso que sea un personal de planta se desea saber el área al cual pertenece. De esta forma podrá ver qué área o que laboratorio necesita más personal.

[nombre, apellido, teléfono, DNI, laboratorio, área]

| SELECT  emp.nombre,  emp.apellido,  emp.telefono,  emp.DNI,  lab.uso Laboratorio,  ar.area Area FROM  empleado emp  LEFT JOIN  investigador inv ON inv.id\_empleado = emp.id\_empleado  LEFT JOIN  personal\_planta pp ON pp.id\_empleado = emp.id\_empleado  LEFT JOIN  laboratorio lab ON lab.id\_laboratorio = inv.id\_laboratorio  LEFT JOIN  area ar ON ar.id\_area = pp.id\_area ORDER BY ar.id\_area; |
| --- |

* La empresa desea tener un registro detallado de todo su personal. Se desea conocer toda la información de los empleados, id, datos, su sueldo, rol, titulo, DNI, central sindical (si es que se encuentra afiliado a alguna) y fecha de contratación.

[id, DNI, nombre, apellido, telefono, central\_sindical, rol, titulo, fecha\_contratacion, sueldo]

| SELECT  e.id\_empleado ID,  e.DNI DNI,  e.nombre NOMBRE,  e.apellido APELLIDO,  e.telefono TELEFONO,  cs.central\_sindical CENTRAL\_SINDICAL,  ro.rol ROL,  t.titulo TITULO,  co.fecha FECHA\_CONTRATACION,  ro.sueldo SUELDO FROM  empleado e  LEFT JOIN  central\_sindical cs ON e.id\_central\_sindical = cs.id\_central\_sindical  INNER JOIN  responsabilidad re ON e.id\_empleado = re.id\_empleado  INNER JOIN  rol ro ON re.id\_rol = ro.id\_rol  INNER JOIN  titulacion ti ON e.id\_empleado = ti.id\_empleado  INNER JOIN  titulo t ON ti.id\_titulo = t.id\_titulo  INNER JOIN  contrato co ON e.id\_empleado = co.id\_empleado ORDER BY e.id\_empleado; |
| --- |

* Recientemente se ha extraviado un artículo importante del laboratorio con id 2. Se desea ver la información de los investigadores que trabajan en dicho laboratorio y sus ingresos y egresos al laboratorio para encontrar posibles sospechosos, ya que una de las políticas de la empresa consiste en que los investigadores no pueden estar en el laboratorio fuera del horario de trabajo asignado. Gracias a un estudio confiable se sabe que la desaparición de dicho artículo se dio entre el 17 y 20 de enero del presente año. Solicita que este ordenado por fecha.

[fecha\_hora, id\_empleado, DNI, estado (puede ser ingresa si vale 1 o egresa si vale 0)]

| SELECT  ie.fecha\_hora,  e.id\_empleado,  e.DNI,  ie.ingreso estado FROM  ingresa\_egresa ie  INNER JOIN  empleado e ON ie.id\_empleado = e.id\_empleado WHERE  ie.id\_laboratorio = 2  AND  (DAY(ie.fecha\_hora) BETWEEN 17 AND 20)  AND  MONTH(ie.fecha\_hora) = 1 ORDER BY ie.fecha\_hora; |
| --- |

* La empresa desea analizar la información de aquellas facturas de venta que fueron anuladas, para de esta forma evaluar el motivo y encontrar alguna forma de que se disminuya esta posibilidad. Se desea conocer la información de aquellas facturas de venta anuladas, la fecha, el id de factura, el cliente y el monto de la misma. Se lo solicita ordenado por fecha.

[fecha, id\_factura\_venta, cliente, monto]

| SELECT  fcv.fecha,  fcv.id\_factura\_venta,  cliente,  SUM(cantidad\*precio\_unitario) Monto  FROM  factura\_cabecera\_venta fcv   INNER JOIN  cliente cl ON cl.id\_cliente = fcv.id\_cliente   INNER JOIN  factura\_detalle\_venta fdv ON fcv.id\_factura\_venta = fdv.id\_factura\_venta  WHERE  anulada = 1  GROUP BY fcv.id\_factura\_venta  ORDER BY fcv.fecha; |
| --- |

* La empresa quiere ver su stock de productos y materiales en su sucursal. Desea conocer la cantidad de materiales disponibles en el almacén por tipo de material, al igual que los productos por tipo de producto.

[tipo, cantidad, esMaterial(el cual tendrá valor 0 si se trata de un producto y 1 si es material)]

| (SELECT  tp.tipo,  SUM(prod.cantidad) cantidad,  0 esMaterial FROM  tipo\_producto tp  INNER JOIN  producto prod ON prod.id\_tipo\_producto = tp.id\_tipo\_producto GROUP BY tp.id\_tipo\_producto) UNION (SELECT  tm.tipo,  SUM(mat.cantidad) cantidad,  1 esMaterial FROM  tipo\_material tm  INNER JOIN  material mat ON mat.id\_tipo\_material = tm.id\_tipo\_material GROUP BY tm.id\_tipo\_material); |
| --- |

* La empresa está buscando renovar su stock, para esto, desea conocer información de los tres productos y materiales que menos cantidad se disponga. De esta manera fabricar urgentemente aquellos productos y comprar dichos materiales.

[tipo, objeto, cantidad, esProducto(el cual tendrá valor 1 si es un producto y 0 si es un material)]

| (SELECT  tipo,  producto objeto,  cantidad,  1 esProducto  FROM  producto pr   INNER JOIN  tipo\_producto tp ON pr.id\_tipo\_producto = tp.id\_tipo\_producto  ORDER BY cantidad LIMIT 3)  UNION (SELECT  tipo,  material objeto,  cantidad,  0 esProducto  FROM  material mat   INNER JOIN  tipo\_material tm ON mat.id\_tipo\_material = tm.id\_tipo\_material  ORDER BY cantidad LIMIT 3); |
| --- |

* La empresa necesita conocer el rendimiento de las ventas, para esto quiere comparar las ventas obtenidas con el promedio de todas sus ventas y ver si vendió por debajo o por encima de este. Para esto desea conocer, ordenados por fechas, los importes obtenidos esa fecha, el importe promedio y la diferencia entre el importe de la fecha correspondiente y el importe promedio.

[fecha, importe, promedio, diferencia]

| SELECT  fcv.fecha Fecha,  SUM(fdv.cantidad \* fdv.precio\_unitario) Importe,  @prom := (SELECT  TRUNCATE(AVG(fdv.cantidad \* fdv.precio\_unitario), 3) PromedioVentas  FROM  factura\_detalle\_venta fdv  INNER JOIN  factura\_cabecera\_venta fcv ON fdv.id\_factura\_venta = fcv.id\_factura\_venta  WHERE  fcv.anulada = 0) Promedio,  TRUNCATE((SUM(fdv.cantidad \* fdv.precio\_unitario) - @prom), 3) Diferencia FROM  factura\_detalle\_venta fdv  INNER JOIN  factura\_cabecera\_venta fcv ON fdv.id\_factura\_venta = fcv.id\_factura\_venta WHERE  fcv.anulada = 0 GROUP BY Fecha ORDER BY fcv.fecha; |
| --- |

| WITH RECURSIVE videos (id\_canal, id\_root, nombre\_canal, id\_video, cant\_me\_gusta, cant\_no\_me\_gusta) AS (  SELECT  c.id\_canal,  c.id\_canal,  c.nombre,  v.id\_video,  v.cant\_me\_gusta,  v.cant\_no\_me\_gusta  FROM  colaboracion col  INNER JOIN  video v ON col.id\_video = v.id\_video  RIGHT JOIN  canal c ON col.id\_canal\_subido = c.id\_canal  WHERE  c.id\_canal\_padre IS NULL   UNION DISTINCT   SELECT  c.id\_canal,  vid.id\_root,  c.nombre,  v.id\_video,  v.cant\_me\_gusta,  v.cant\_no\_me\_gusta  FROM  colaboracion col  INNER JOIN  video v ON col.id\_video = v.id\_video  RIGHT JOIN  canal c ON col.id\_canal\_subido = c.id\_canal  INNER JOIN  videos vid ON c.id\_canal\_padre = vid.id\_canal ) SELECT  id\_root id\_canal,  nombre\_canal,  SUM(IF(id\_video IS NULL, 0, 1)) cant\_videos,  COALESCE(SUM(cant\_me\_gusta), 0) cant\_me\_gusta,  COALESCE(SUM(cant\_no\_me\_gusta), 0) cant\_no\_me\_gusta FROM  videos GROUP BY id\_root; |
| --- |