Data Base

Universidad Industrial de Santander

Diseño físico de la Base de Datos

El diseño físico de la base de datos optimiza el rendimiento a la vez que asegura la integridad de los datos al evitar repeticiones innecesarias de datos.

Durante el diseño físico, se transforman las entidades en tablas, las instancias en filas y los atributos en columnas.

- Convertir entidades en tablas físicas
- Utilizar para las columnas de las tablas físicas
- Columnas de las tablas deben definirse como claves
- Indices deben definirse en las tablas
- Vistas deben definirse en las tablas
- Desnormalizar las tablas

Elegir un SGBD

Los tres factores más influyentes a la hora de elegir un servidor de bases de datos son:

- El tipo de datos
- Aplicaciones utilizadas
- Rendimiento requerido.

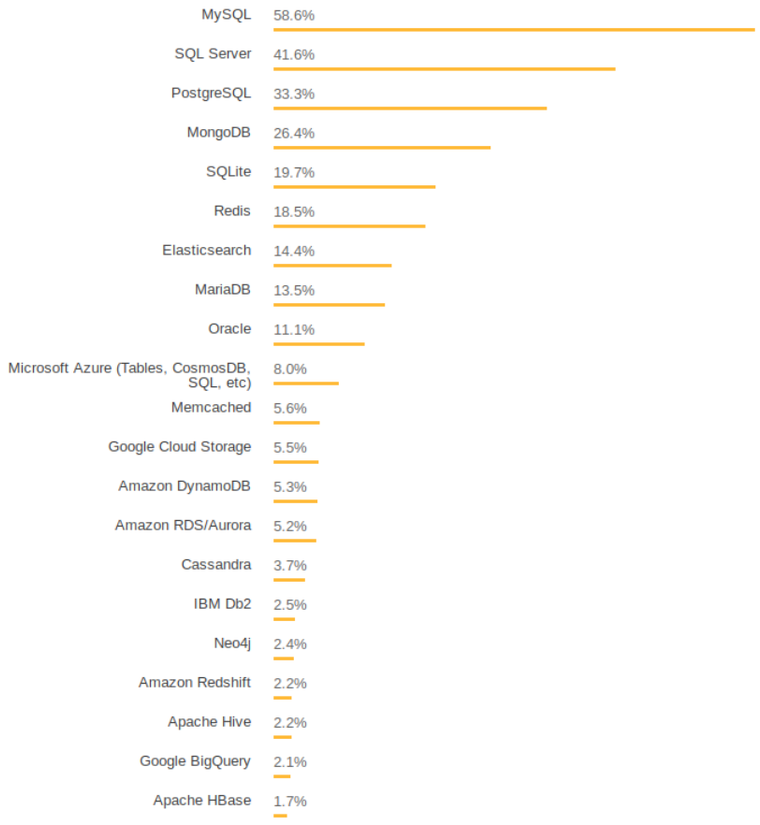
El criterio que mas pesa en los desarrolladores.

La integración con las tecnologías usadas.

Sin dejar a un lado el rendimiento.

Elegir un SGBD

Ejemplos de SGBD. Algunos de los mas Importantes SGBD son:



*Tomado stackoverflow

Elegir un SGBD

Data Base	Características
MySQL	Compatibilidad con SQL. Arquitectura cliente/servidor. Procedimientos almacenados. Soporte multiplataforma. Soporte de Unicode. Consulta de caché. Soporte SSL.
SQL Server	Admite una amplia variedad de aplicaciones de procesamiento de transacciones. SQL está vinculado a Transact-SQL (T-SQL). Visualización de datos e informes en dispositivos móviles. CompatibiliDad con nube híbrida. Escalabilidad y seguridad.
PostgreSQL	Tipos definidos por el usuario Herencia de tablas. Extensibilidad. Mecanismo de bloqueo sofisticado. Clave foránea de integridad referencial. Integridad de datos. Vistas, reglas, subconsultas. Control de concurrencia multi-versión (MVCC).
SQLite	Configuración cero Sin servidor Archivo de base de datos único Compacto Registro de longitud variable Código fuente legible.
MariaDB	Admite API de JSON. Replicación de datos en paralelo. Múltiples motores de almacenamiento. Notablemente escalable. Amplia selección de motores de almacenamiento. Utiliza un lenguaje de consulta estándar y popular. Velocidad y alta seguridad.
Oracle	Totalmente escalable. Inteligencia de negocios. Agrupamiento Gestión de contenidos. Servicios de localización. Gestión del servidor. Inteligencia de negocios. Alto rendimiento, seguridad y análisis

Data Base

Pasos para crear.

1) Instalar el motor.

PostgresSql.

http://www.postgresqltutorial.com/install-postgresql/

MySql.

https://dev.mysql.com/doc/refman/8.0/en/installing.html

Oracle

https://docs.oracle.com/en/database/oracle/oracle-database/19/install-and-upgrade.html

SqLite

https://www.sqlitetutorial.net/download-install-sqlite/

Data Base

Pasos para crear.

2)Escoger Un software de administración de bases de datos

PostgresSql.

PgAdmin

MySql.

MySQL Workbench

Oracle

Toad for Oracle

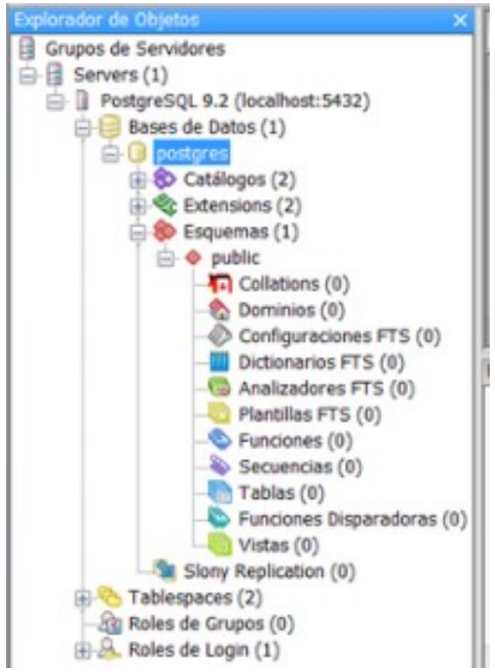
SqLite

DB Browser for SQLite

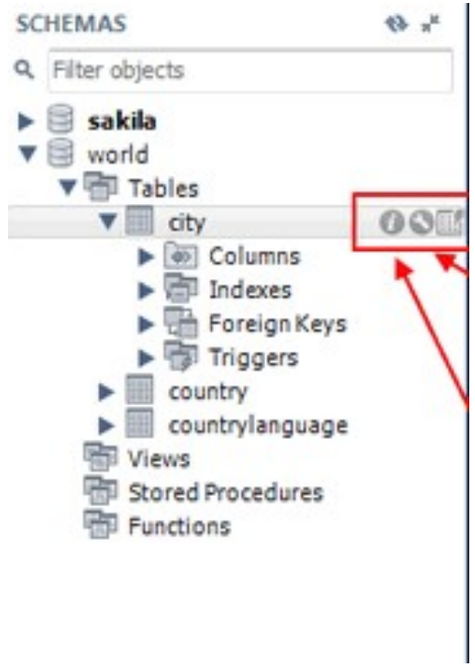
Pasos para crear.

3)Crear la Base de datos.

Postgres



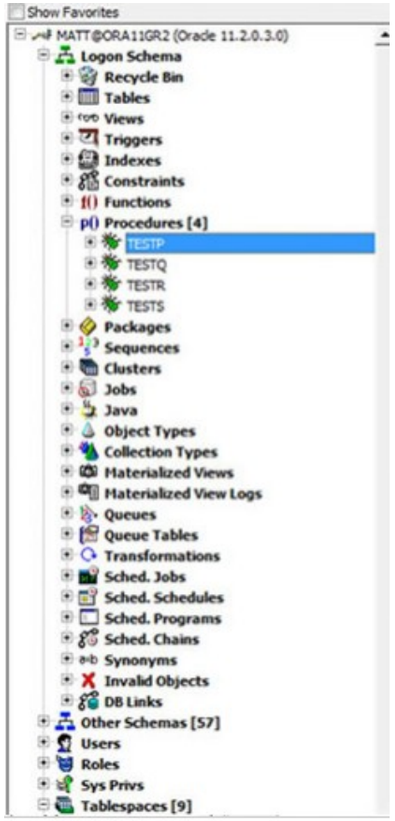
Mysql



Pasos para crear.

3)Crear la Base de datos.

Oracle



Pasos para crear.

3)Crear la Base de datos.

Postgres

CREATE DATABASE DataBaseName;

Mysql

CREATE DATABASE menagerie;

Pasos para crear.

3)Crear la Base de datos.

Postgres

CREATE DATABASE DataBaseName;

Mysql

CREATE DATABASE menagerie;

Oracle

CREATE DATABASE menagerie;

pero revisar la forma correcta

https://docs.oracle.com/cd/B28359_01/server.111/b28310/create003.htm#ADMIN11074

Pasos para crear.

4) Crear la Esquema

Postgres

```
CREATE SCHEMA myschema;
```

Oracle

```
CREATE SCHEMA AUTHORIZATION schema_name
[create_table_statement]
[create_view_statement]
[grant_statement];
```

Pasos para crear.

4)Crear la Table

```
CREATE TABLE table_name (
   column_name TYPE column_constraint,
   table_constraint table_constraint
) INHERITS existing_table_name;
```

Para crear una tabla, debe proporcionar un nombre de la entidad y no debe existir otra con el mismo nombre.

Pasos para crear.

4)Crear la Table

column_name **TYPE** column_constraint.

Tipos de datos :Algunos de los tipos de datos básicos de SQL son estos dependen del **SGBD.**

Varchar: Recibe cadena de palabras compuestas de letras, números y caracteres especiales.

int es el principal tipo de datos de valores enteros de SQL Server. Con números enteros con o sin signo

Date: una fecha de calendario que contiene el año (de cuatro cifras), el mes y el día.

Time: La hora del día en horas minutos segundos (el valor predeterminado es 0).

Pasos para crear.

4)Crear la Table

column_name **TYPE** column_constraint.

Postgres

https://www.postgresql.org/docs/9.2/datatype.html

Mysql

http://www.mysqltutorial.org/mysql-data-types.aspx

Oracle

https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm#CNCPT1828

Pasos para crear.

4)Crear la Table

column_name **TYPE** auto incremento.

Postgres

id SERIAL PRIMARY KEY,

Mysql

id MEDIUMINT NOT NULL AUTO_INCREMENT,

Oracle solo desde 12c.

"ID" NUMBER(10,0) GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE NOT NULL ENABLE,

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

los **constraint** de SQL se utilizan para especificar reglas para los datos de una tabla. Las restricciones se utilizan para limitar el tipo de datos que pueden entrar en una columna. Esto asegura la exactitud y fiabilidad de los datos de la tabla. Si hay alguna violación entre la restricción y la acción de los datos, se aborta la acción.

Las restricciones pueden estar al mismo nivel de columna o nivel de tabla. restricciones a nivel de la columna se aplican a una columna, y las restricciones a nivel de tabla se aplican a toda la tabla.

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

NOT NULL - La columna no puede tener un valor Nulos

UNIQUE - todos los valores de una columna son diferentes

PRIMARY KEY -Identifica de forma exclusiva cada fila de una tabla

FOREIGN KEY - Identifica unívocamente de una fila / registro de otra tabla.

CHECK - Se asegura de que todos los valores en una columna satisface una condición específica

DEFAULT - Establece un valor por defecto para una columna cuando no se especifica ningún valor

INDEX - Se utiliza para crear y recuperar datos de la base de datos muy rápidamente

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

NOT NULL - La columna no puede tener un valor Nulos

```
CREATE TABLE Persona (
    ID int NOT NULL,
    Apellido varchar(255) NOT NULL,
    Nomre varchar(255) NOT NULL,
    Edad int
);
```

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

UNIQUE - todos los valores de una columna son diferentes

```
CREATE TABLE Persons (
    ID int NOT NULL UNIQUE,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int
);
```

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

PRIMARY KEY -Identifica de forma exclusiva cada fila de una tabla

```
CREATE TABLE Persons (
    ID int NOT NULL PRIMARY KEY,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int
);
```

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

PRIMARY KEY -Identifica de forma exclusiva cada fila de una tabla

```
CREATE TABLE Persona (
    ID int NOT NULL,
    Apellido varchar(255) NOT NULL,
    Nombre varchar(255),
    Edad int,
    CONSTRAINT PK_Person PRIMARY KEY (ID,Apellido)
);
```

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

FOREIGN KEY - Identifica unívocamente de una fila / registro de otra tabla.

```
CREATE TABLE Orders (
    OrderID int NOT NULL PRIMARY KEY,
    OrderNumber int NOT NULL,
    PersonID int FOREIGN KEY REFERENCES
Persons(PersonID)
);
```

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

FOREIGN KEY - Identifica unívocamente de una fila / registro de otra tabla.

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES
Persons(PersonID)
);
```

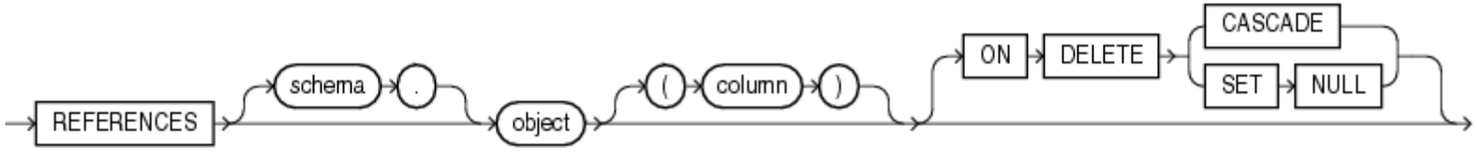
Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

FOREIGN KEY - Identifica unívocamente de una fila / registro de otra tabla.



ON DELETE CASCADE eliminar valores de clave externa dependientes.

ON DELETE SET NULL si desea que convertir los valores de clave externa a NULL.

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

CHECK - Se asegura de que todos los valores en una columna satisface una condición específica utilizando una expresión booleana para evaluar los valores antes de inserción o actualización a la columna

```
CREATE TABLE Empleado (
   id serial PRIMARY KEY,
   Nombre VARCHAR (50),
   Apellido VARCHAR (50),
  Fecha_Nacimineto DATE CHECK (Fecha_Nacimineto > '1900-01-01'),
   Fecha_entrada DATE CHECK (Fecha_entrada>Fecha_Nacimineto ),
   Salario CHECK(salario > 0)
);
```

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

Constraints mas usadas en SQL:

DEFAULT - Establece un valor por defecto para una columna cuando no se especifica ningún valor.

```
CREATE TABLE Persona (
    ID int NOT NULL,
    Apellido varchar(255) NOT NULL,
    Nombre varchar(255),
    edad int,
    Ciudad varchar(255) DEFAULT 'Bucaramanga'
);
```

Pasos para crear.

4)Crear la Table

column_name TYPE column_constraint.

5)Constraints mas usadas en SQL:

INDEX - Se utiliza para crear y recuperar datos de la base de datos muy rápidamente

```
CREATE TABLE Libreria (
        ISBN_NO varchar(15) NOT NULL,
        Autor varchar(40)  ,
        Editorial varchar(40)  ,
        Precio float  ,
        PRIMARY KEY (ISBN_NO),
        INDEX SHORT_DESC_IND (Autor, Editorial)
);
```

Pasos para crear.

4)Crear la Table

```
CREATE TABLE table_name (
   column_name TYPE column_constraint,
   table_constraint table_constraint
) INHERITS existing_table_name;
```

La cláusula opcional INHERITS especifica una colección de nombres de tabla de las cuales esta tabla hereda todos los campos. Permite automáticamente a la tabla creada heredar funciones de las tablas superiores a ella en la jerarquía de herencia.

Pasos para crear.

4)Crear la Table otra forma

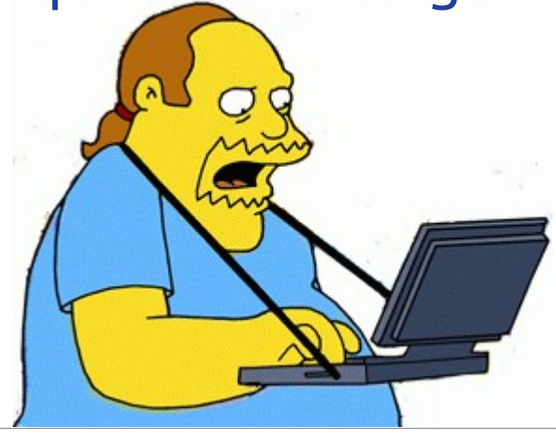
```
CREATE TABLE table_name  AS
SELECT customername, contactname
FROM customers;
```

Pasos para crear.

4) Ejemplo

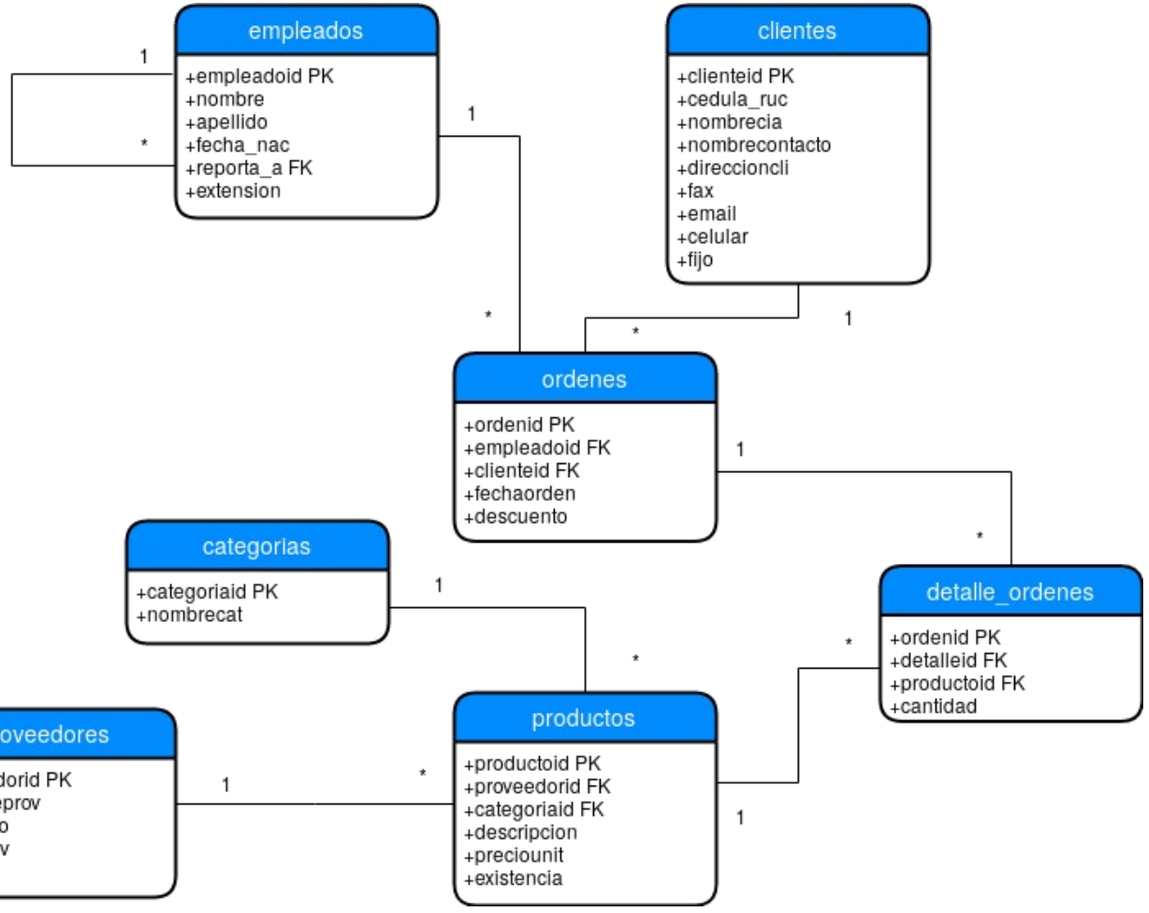
La empresa de ventas de productos de Geek desea sistematizarse , para esto desea gestionar los empleados y a que empleados superior deben reportar.

Los empleados serán los encargados de generar las ordenes de los clientes los cuales contendrán lo productos los cuales están categorizados y tienen un proveedor que se encarga de proporcionarlos al almacén.



Pasos para crear.

5)Ejemplo



Pasos para crear.

5) Ejemplo

Postgres

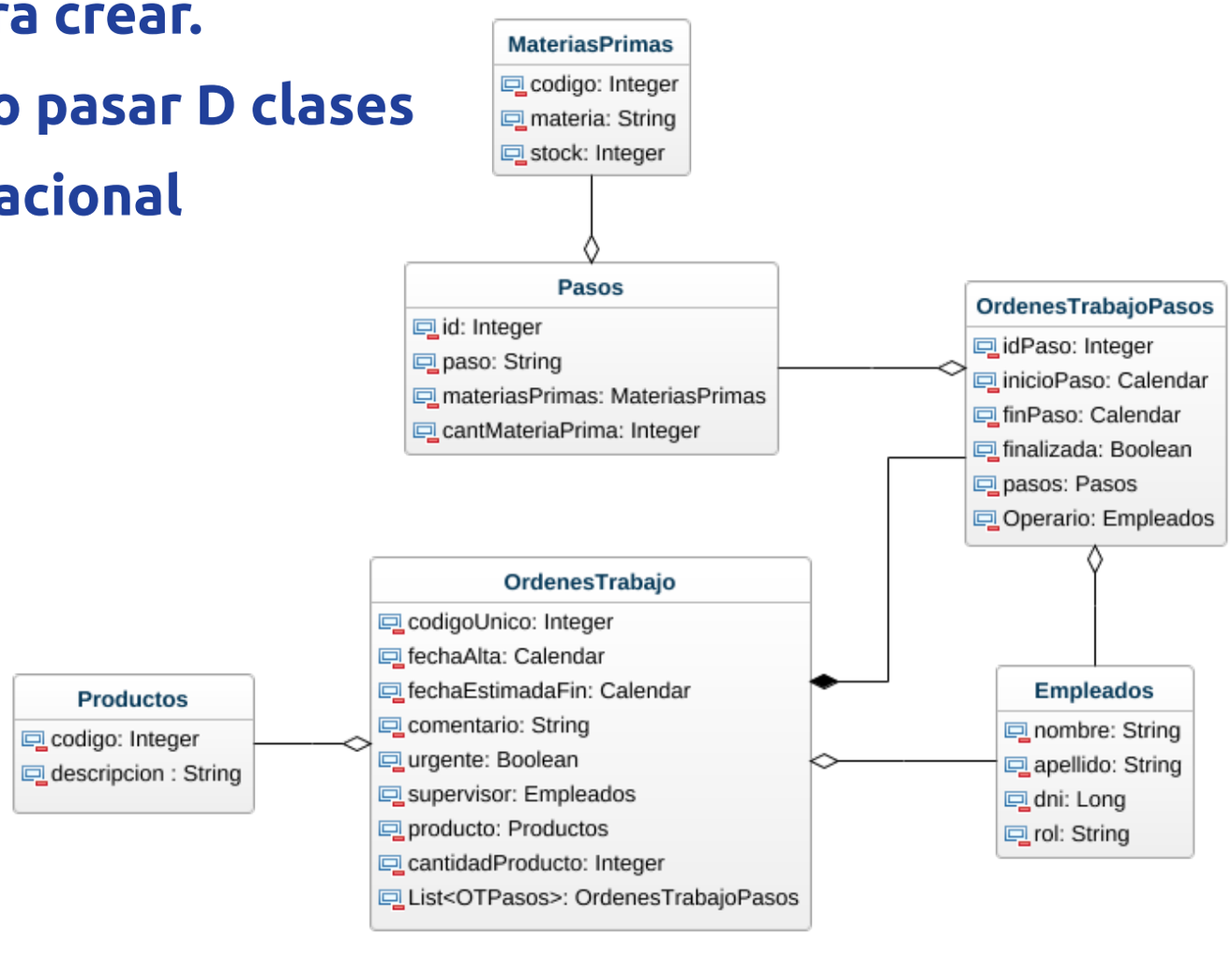
https://drive.google.com/uc?id=1sMNxhDDoJBbqPxLKCSOjq4I9xrIIAw8U&export=download

Oracle

https://drive.google.com/uc?id=1iLs1DTE91wKyad9MjtwJ4dctwLhVQIGG&export=download

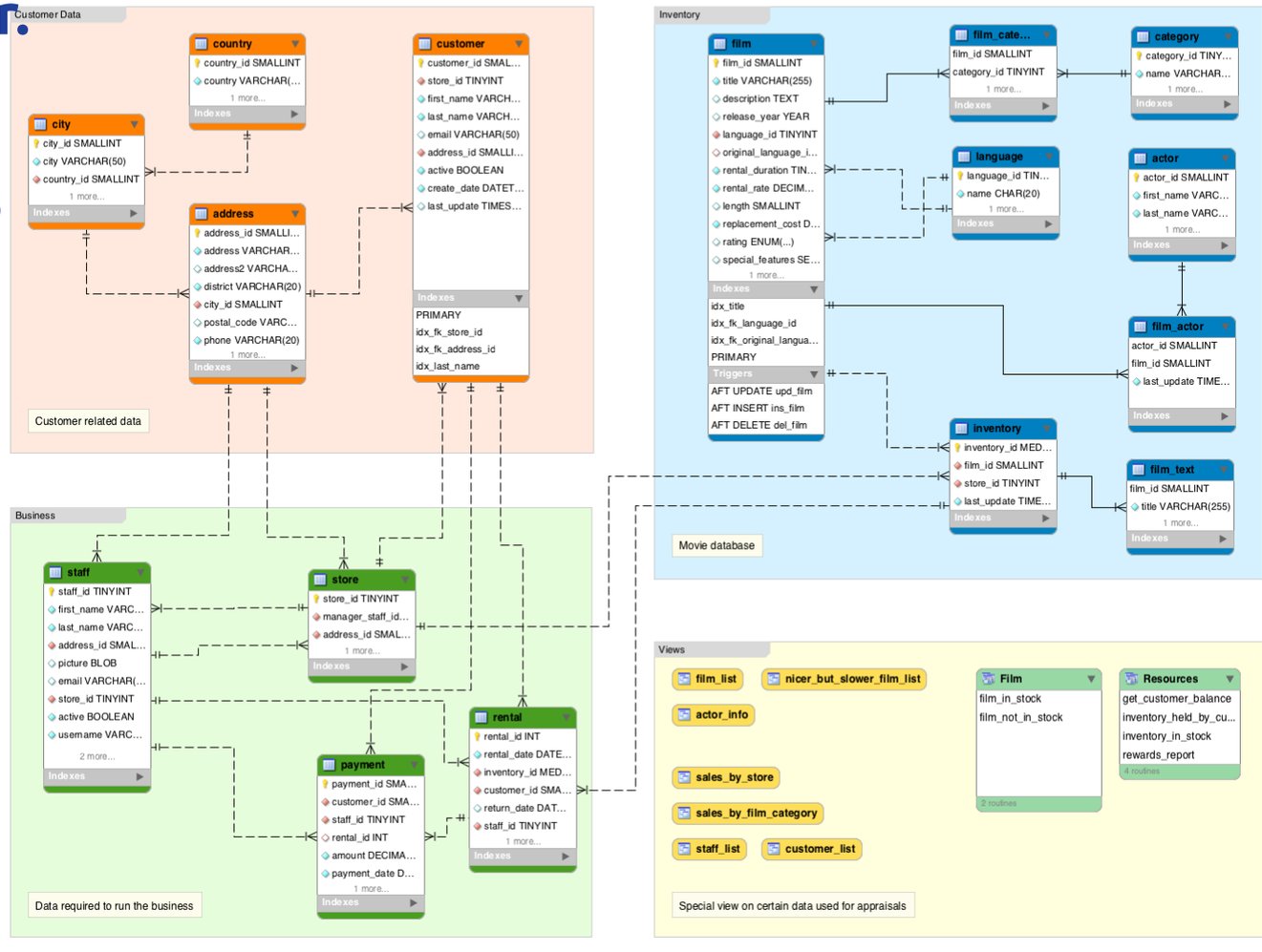
Pasos para crear.

5) Ejemplo pasar D clases a D Relacional



Pasos para crear:

5) Ejemplo por esquemas



Create Table compilado.

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ....
);
```

NOT NULL - Valores nulos.

UNIQUE – Valores únicos en la columna.

PRIMARY KEY – Combinación de NOT NULL and UNIQUE.

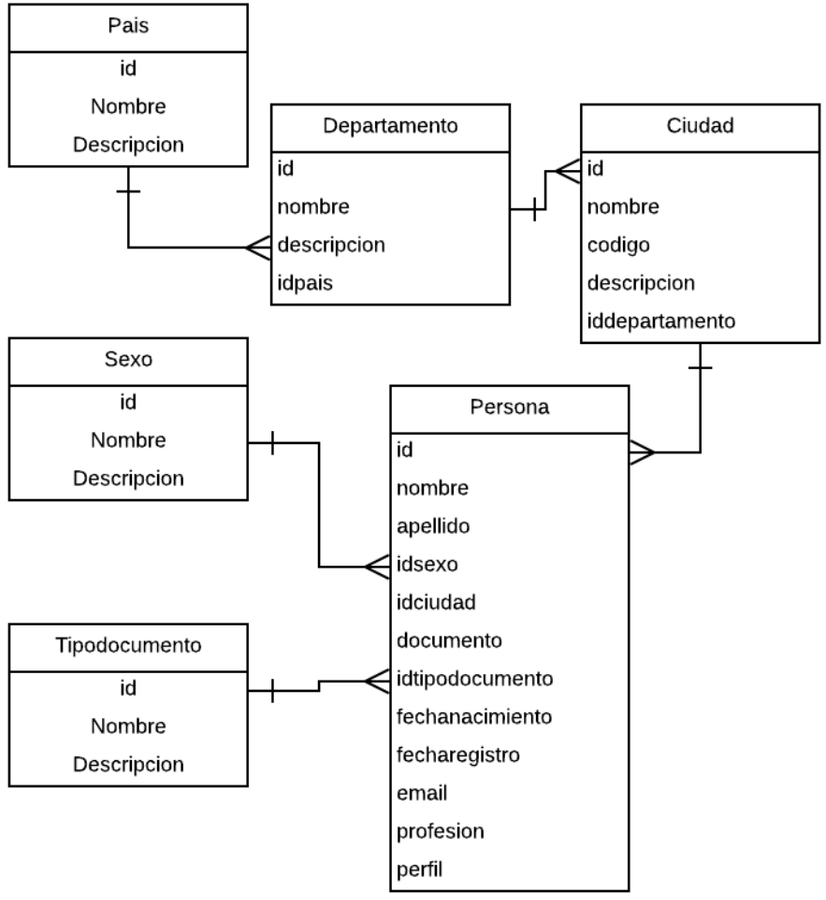
FOREIGN KEY – Registro que se encuentra en otra tabla.

CHECK – La columna satisfagan una condición específica.

DEFAULT -Valor predeterminado cuando no se ningún valor.

INDEX - Recuperar datos de la base de datos muy rápidamente.

Create Table compilado.



https://drive.google.com/uc?id=13l2xf3LBwGCW-0wwWKZgocIjGsBt3wP-&export=download

Create Table compilado.

```
CREATE TABLE Pedido.PERSONA(
ID int NOT NULL,
NOMBRE char(10) NOT NULL,
APELLIDO char(30) NOT NULL,
IDSEXO  int NOT NULL REFERENCES Academico.SEXO(id),
IDCIUDAD int NOT NULL REFERENCES Academico.CIUDAD(id),
DOCUMENTO char(50) NOT NULL,
IDTIPODOCUMENTO  int NOT NULL REFERENCES Academico.TIPODOCUMENTO(id),
FECHANACIMIENTO date NULL CHECK (FECHANACIMIENTO > '1900-01-01'),
FECHAREGISTRO date NOT NULL DEFAULT  Now() ,
email char (355) UNIQUE NOT NULL,
PROFESION char(12) NULL,
PERFIL char(120) NULL,
CONSTRAINT PK_PERSONA PRIMARY KEY
(ID) );
```

Modificar tablas.

Agregar y eliminar columnas

La sentencia alter table permite una amplia gama de formas de modificar una tabla.

agregar una columna, podemos usar la sintaxis siguiente:

```
alter table personal add capital int not null-> after nom;
```

Las columnas no deseadas pueden eliminarse con la opción drop.

```
alter table personal drop pasatiempo;
```

Modificar tablas.

Modificar Columnas

La modificación de una columna con la opción modify es parecida a volver a definirla.

```
ALTER TABLE nombre_tabla ALTER COLUMN nombre_columna
SET DATA TYPE tipo_de_dato;
```

cambiar con esta sintaxis es el nombre de la columna.

```
ALTER TABLE nombre_tabla RENAME COLUMN
actual_nombre_columna TO nuevo_nombre_columna;
```

A demas de esto podemos copiar tablas y Tablas temporales (temporary)