PROGRAMACION AVANZADA DE BASE DE DATOS

Sesión 01

INSTRUCTOR: JOSE FELIPE LEON CABEL
Email: jfleonc@isil.pe

Instructor: José León Cabel

Perfil profesional: Analista desarrollador senior, especialista en implementación de sistemas de gestión empresarial bajo herramientas Microsoft. 32 años de experiencia en el desarrollo, implementación e instrucción en tecnologías de información

E mail : jfleonc@isil.pe

Horario:

NRC 2479 : Viernes 7:00- 9:50

NRC 2478 : Lunes 7:00- 9:50

Sistema de Evaluación

ESQUEMA DE EVALUACIÓN			
Evaluación permanente	(EP)	40%	Sesiones
Evaluación permanente 1	(EP1)		4
Evaluación permanente 2	(EP2)		6
Evaluación permanente 3	(EP3)		11
Evaluación permanente 4	(EP4)		14
Evaluación parcial	(EV.PARCIAL)	30%	8
Evaluación final	(EV. FINAL)	30%	16

Modo de evaluación:

- Evaluación permanente 1 -> Avance Proyecto 1
- Evaluación permanente 2 -> Practica
- Examen parcial -> Avance Proyecto 2
- Evaluación permanente 3 -> Practica
- Evaluación permanente 4 -> Avance Proyecto 3
- Examen final -> Sustentación final

Enfoque del curso

- El curso de Programación Avanzada de Base de datos, esta enfocado a que el participante continúe el proceso de aprendizaje con respecto al curso anterior, es decir, Diseño y Programación de Base de Datos con el objetivo que este capacitado en la implementación de todos los objetos necesarios para que la base de datos brinde el respaldo necesario a las aplicaciones.
- Se contemplaran casos creación de vistas, procedimientos almacenados, manejo de transacciones, funciones del usuario, paginación de datos y auditoria de cambios en base a disparadores. Así mismo se hará referencia a aspectos básicos de seguridad y de generación y restauración de copias de respaldo de la base de datos.
- Revisemos el Syllabus del curso para conocerlo mejor.

Hablemos del proyecto integrador…

- Aplicando el slogan de ISIL “Aprende haciendo” , la mejor forma de evaluar a los participantes de un curso, mas aun si es de carácter practico como el nuestro, es la de implementar un proyecto real (o lo mas cercano a la realidad).
- Revisemos el documento asociado al proyecto integrador del curso y expliquemos lo referente al mismo.

Link de descarga de SQL Server 2019

- El siguiente es el link para la descarga de la versión Desarrollador o la Express del producto:

https://www.microsoft.com/es-es/sql-server/sql-server-downloads

Recomendaciones…

Nos permitimos darte algunas sugerencias para llevar el curso de mejor manera

- Asiste a las clases en el día y la hora que te matriculaste, y emplea el video como una herramienta de refuerzo a lo aprendido. Recuerda que es un curso de TECNOLOGIA.
- Desarrolla los ejercicios en clase, tal cual te lo pide tu instructor. También desarrolla las guías practicas que se dejaran en cada clase. Recuerda que la practica hace al maestro.
- El instructor NO COMPARTIRA EL CODIGO (salvo casos puntuales) al finalizar la sesión, por lo que es tu obligación desarrollar los ejercicios planteados.

Antes de empezar…

Quisiéramos saber tu respuesta acerca de estas 3 preguntas :

- ¿Acerca de que tema fue el proyecto que desarrollaste en el curso de Diseño y Programación de BD?
- ¿Has desarrollado un aplicativo o formado parte de un equipo de proyecto para hacerlo?
- ¿Que esperas del curso?

SESIÓN 01

1. ARQUITECTURA CLIENTE/SERVIDOR

Temario

1. Arquitectura Cliente/Servidor
2. Revisión de conceptos
   - Creación de una base de datos.
   - Creación de tablas.
   - Definición de las llaves primarias (PK) y foráneas (FK)
   - Implementación de constraints : Check, Default y Unique
   - Adición de registros con Insert.

/ OBJETIVOS

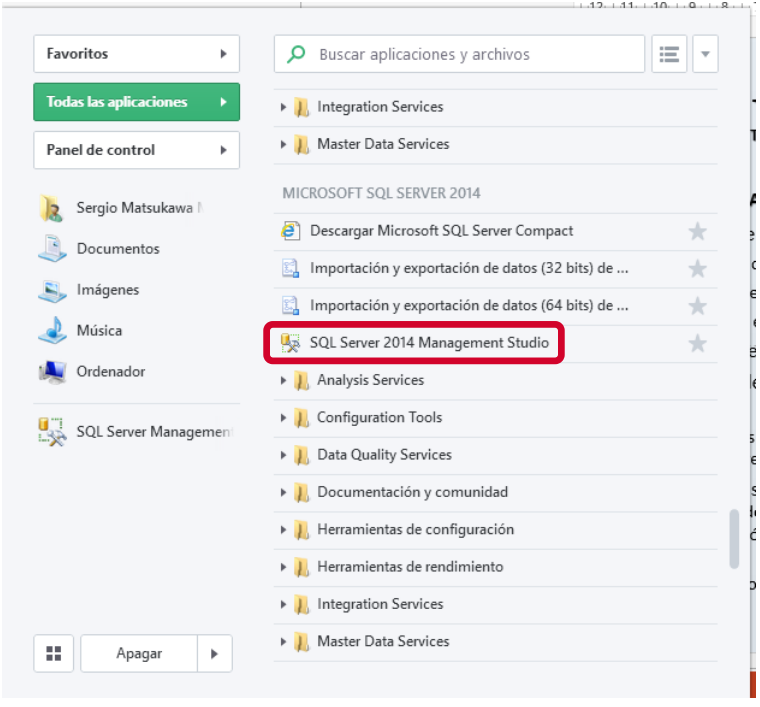
Dispositivos cliente ejecutando aplicaciones cliente

Requerimientos

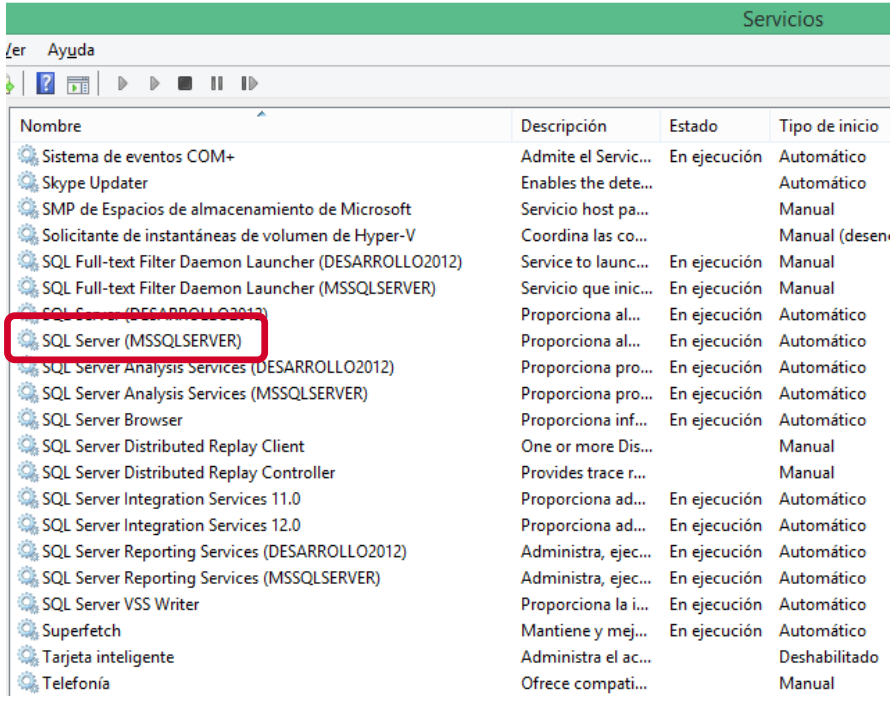
Resultados

Servidores

/ OBJETIVOS



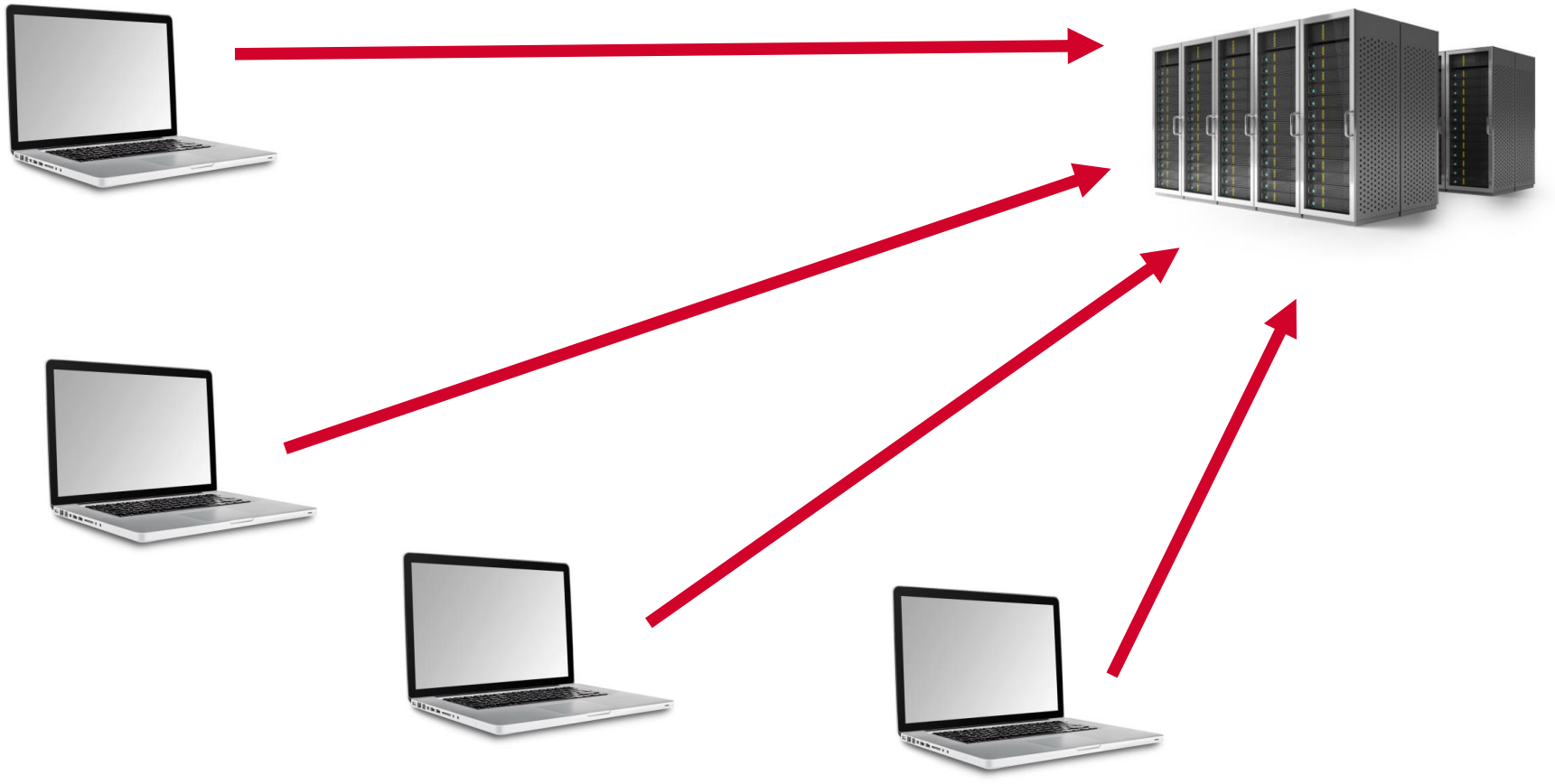
Aplicación cliente en el Menú Inicio de Windows



Aplicación servidor en la ventana Servicios de Windows

SQL
Structured Query Language
Lenguaje de Consulta Estructurado

/ OBJETIVOS

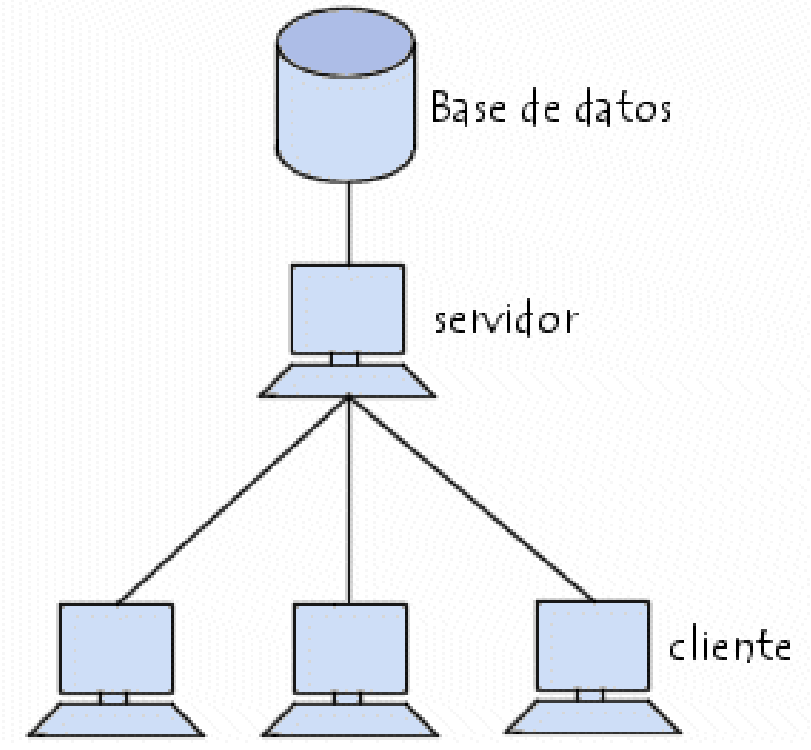


/ ARQUITECTURA CLIENTE/SERVIDOR

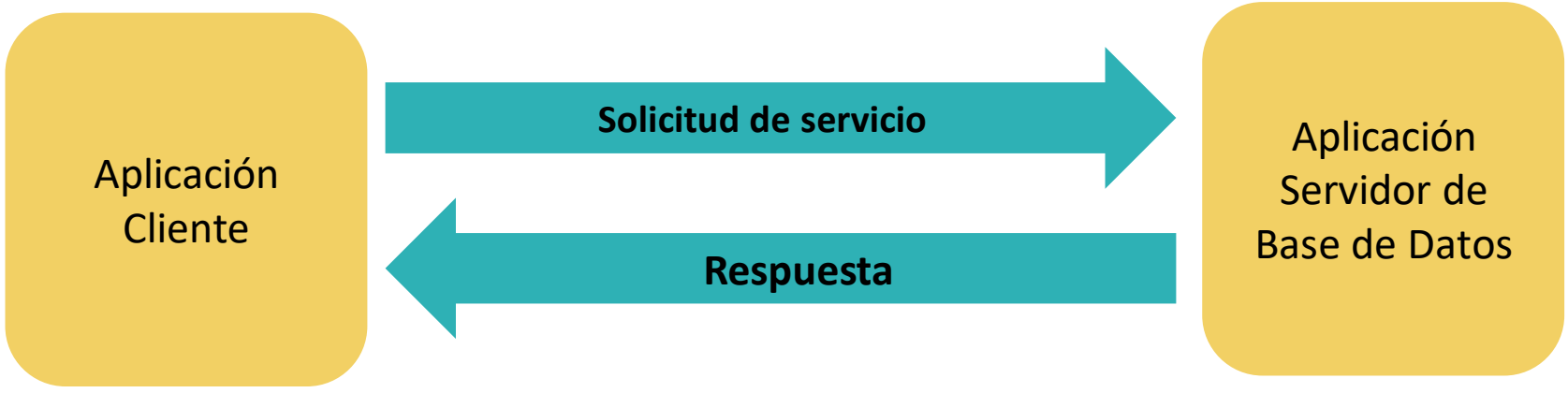
- Servidor: rol que desempeña un equipo ofreciendo un conjunto de servicios a los clientes, tales como manejo de archivos, impresión, páginas web, direccionamiento de correo electrónico, actualización de BD y control de acceso.

- Cliente: rol que desempeña un equipo demandando servicios de los servidores, pero también puede realizar procesamiento local, tales como desplegar páginas web, mostrar ventanas y generar correo electrónico.

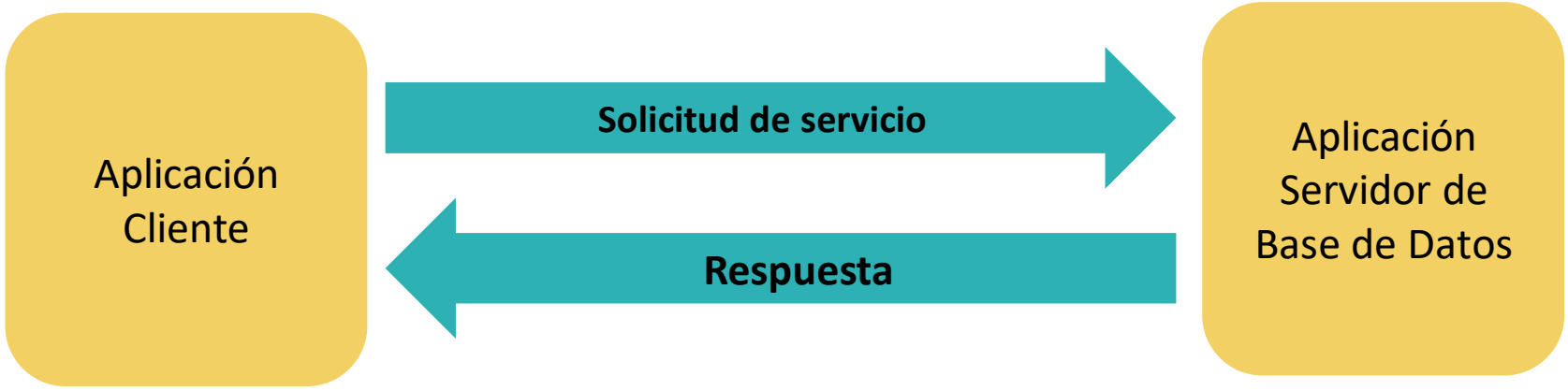
- Eventualmente un mismo equipo puede desempeñar ambos roles.



/ ARQUITECTURA CLIENTE/SERVIDOR



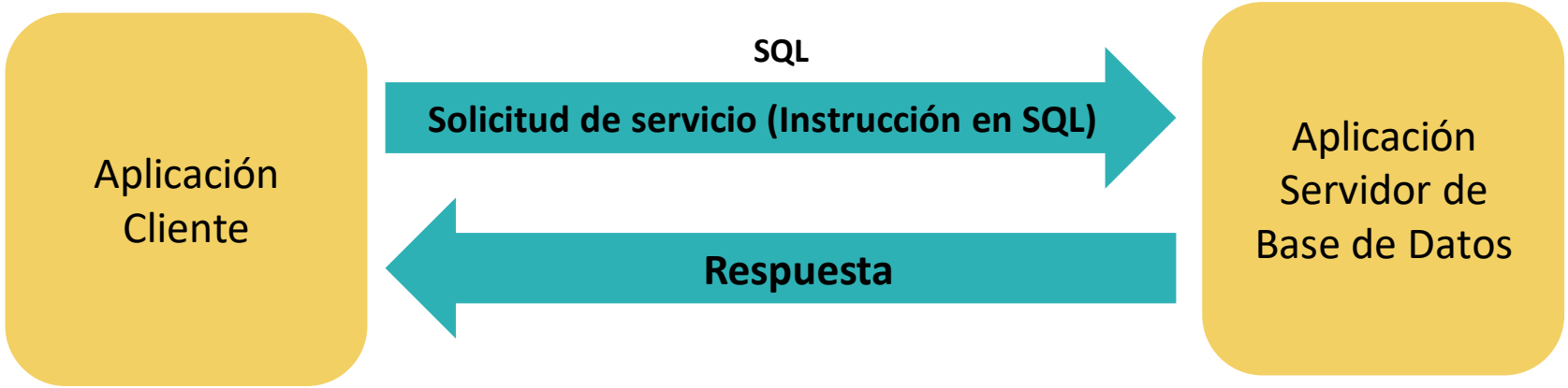
/ ARQUITECTURA CLIENTE/SERVIDOR



Java
JavaScript
Visual Basic
VBA
C++
C#
Python
…

Oracle DB
MS SQL Server
MS Access
DB2
Informix
Sybase DB
…

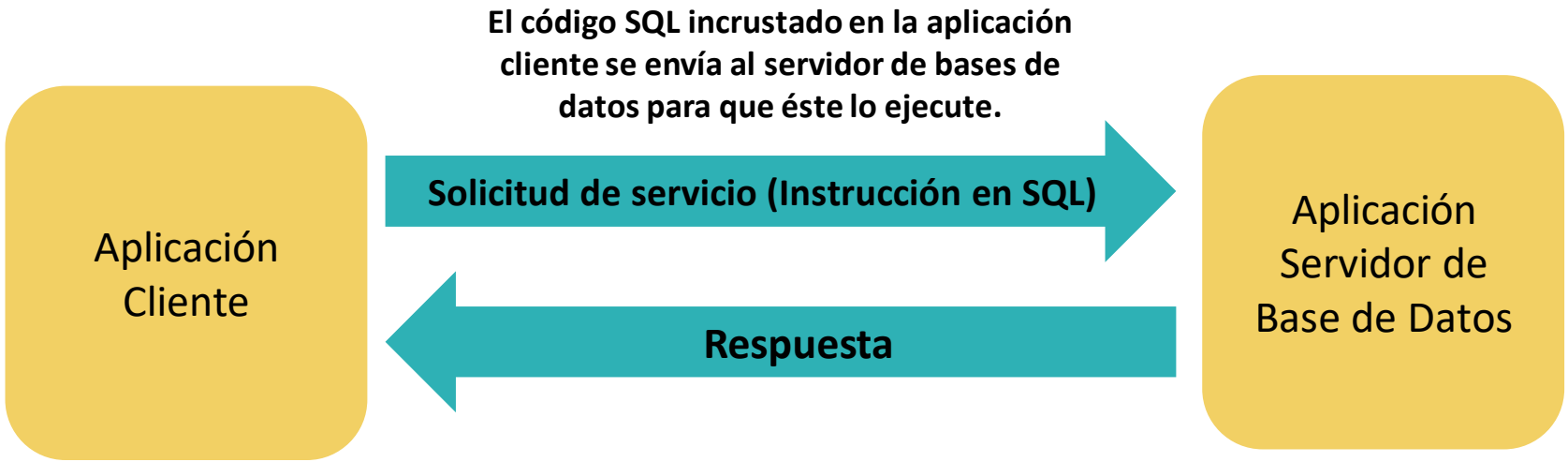
/ ARQUITECTURA CLIENTE/SERVIDOR



Java
JavaScript
Visual Basic
VBA
C++
C#
Python
…

Oracle DB
MS SQL Server
MS Access
DB2
Informix
Sybase DB
…

/ ARQUITECTURA CLIENTE/SERVIDOR



Java
JavaScript
Visual Basic
VBA
C++
C#
Python
…

Oracle DB
MS SQL Server
MS Access
DB2
Informix
Sybase DB
…

SESIÓN 01

2. REVISION DE CONCEPTOS. INTRODUCCIÓN A SQL

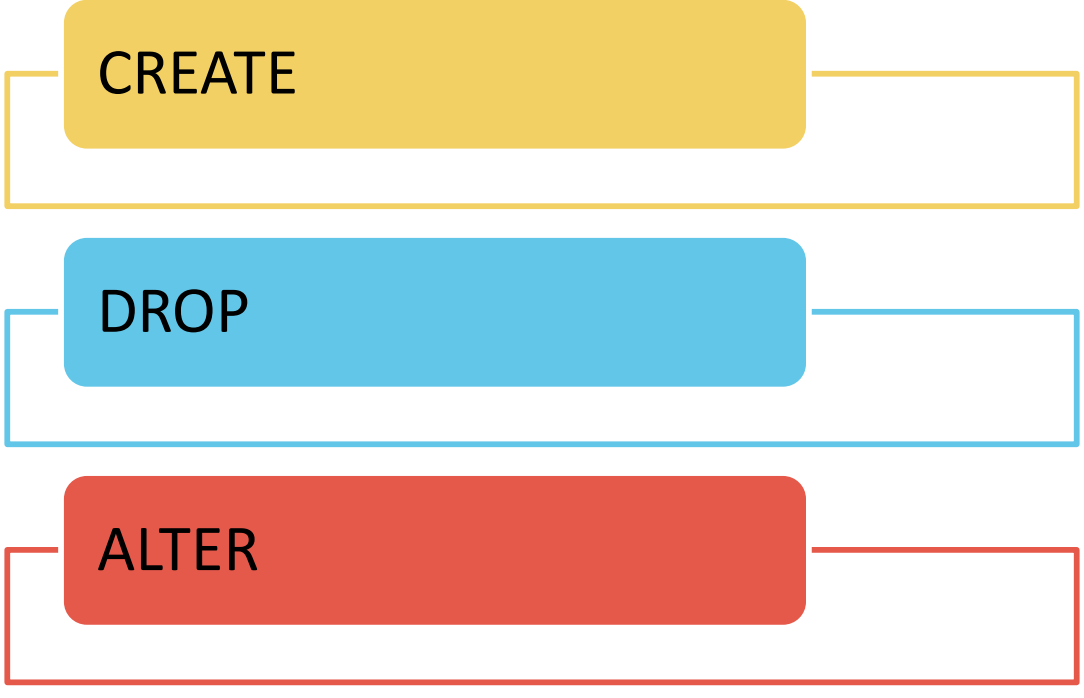
SQL
Structured Query Language
Lenguaje de Consulta Estructurado

RDBMS
Relational DataBase Management System

- SQL (Structured Query Language) es un lenguaje de programación estándar e interactivo para la obtención de información desde una base de datos y para actualizarla.

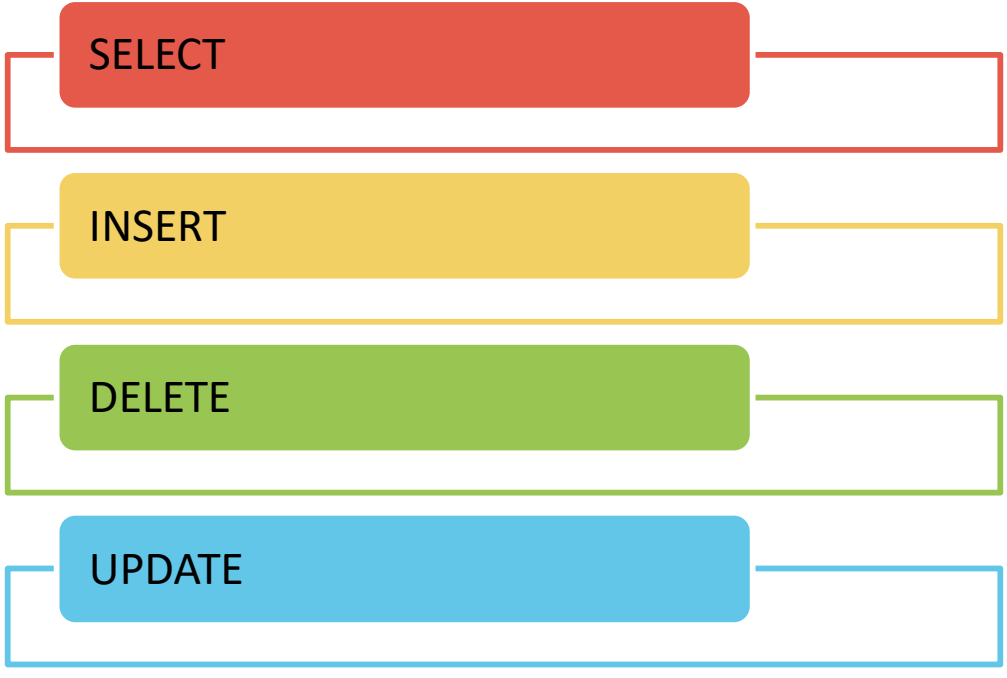
DECLARACIONES DDL
(Data Definition Language)

- Se utilizan para crear y modificar la estructura de las tablas, así como otros objetos de la base de datos.



DECLARACIONES DML
(Data Manipulation Language)

- Permite generar consultas para ordenar, insertar, actualizar, filtrar, agrupar y extraer datos de la base de datos.



/ ACTIVIDAD

OBJETIVO

- Reforzar el procedimiento para iniciar una sesión en un servidor SQL Server utilizando el cliente SQL Server Management Studio.

- Reforzar el uso de la ventana de consultas de SQL Server Management Studio.

TAREA

- Escribir las declaraciones SQL para crear la siguiente tabla en una base de datos de nombre **Tarea1**
  - Nombre de la tabla: **Estudiante.**
  - Columna 1: de nombre **código**, de tipo numérico entero que no permite valores nulos. Clave primaria de la tabla.
  - Columna 2: de nombre **apellidos**, de tipo cadena de longitud variable de hasta 50 caracteres, no permite valores nulos.
  - Columna 3: de nombre **nombre**, de tipo cadena de longitud variable de hasta 35 caracteres, no permite valores nulos.
  - Columna 4: de nombre **fechaNacimiento**, de tipo fecha, permite valores nulos.
  - Columna 5: de nombre **nuevo,** valor booleano que indica si el estudiante es nuevo o no, no permite nulos.
  - Columna 6: de nombre **monto**, de tipo numérico con 2 decimales, permite nulos.

INSTRUCCIONES PARA GUARDAR EL ARCHIVO CON LA SOLUCIÓN DE SU TAREA.

- Con el cursor ubicado en la ventana de consultas que contiene la solución de su tarea haga clic en el menú **Archivo.**
- Clic en **Guardar SQLQueryX.slq como…**
- Seleccione la carpeta destino.
- En **Nombre** digite **Tarea1-suNombre-suApellidoPaterno.**
- Clic en **Guardar**. Se guarda como un archivo con extensión .SQL.
- NOTA: Solo si su instructor lo indique suba el archivo a la plataforma de trabajo.

Creación de clave primaria (PK)

Los constraints de PRIMARY KEY identifican la columna o el conjunto de columnas cuyos valores identifican de forma unívoca cada una de las filas de una tabla.

FORMA 1:

```
CREATE TABLE Tb_Cliente
(
    Cod_cli nvarchar(4) NOT NULL,
    Raz_soc_cli nvarchar(100) NOT NULL,
    Dir_cli nvarchar(50) NULL,
    Tel_cli nvarchar(10) NULL,
    Ruc_cli nchar(11) NOT NULL,
    Id_Ubigeo nchar(6) NULL,
    Tip_cli nvarchar(1) NOT NULL,
    Contacto nvarchar(30) NULL,
    Fec_reg datetime NOT NULL,
    Usu_Registro varchar(20) NULL,
    Fec_Ult_Mod datetime NULL,
    Usu_Ult_Mod varchar(20) NULL,
    Est_cli int NULL,
    PRIMARY KEY
            (Cod_cli ASC)
)
GO
```

FORMA 2:

```
CREATE TABLE Tb_Cliente
(
    Cod_cli nvarchar(4) NOT NULL,
    Raz_soc_cli nvarchar(100) NOT NULL,
    Dir_cli nvarchar(50) NULL,
    Tel_cli nvarchar(10) NULL,
    Ruc_cli nchar(11) NOT NULL,
    Id_Ubigeo nchar(6) NULL,
    Tip_cli nvarchar(1) NOT NULL,
    Contacto nvarchar(30) NULL,
    Fec_reg datetime NOT NULL,
    Usu_Registro varchar(20) NULL,
    Fec_Ult_Mod datetime NULL,
    Usu_Ult_Mod varchar(20) NULL,
    Est_cli int NULL
)
GO

ALTER TABLE Tb_Cliente
    ADD PRIMARY KEY
            (Cod_cli ASC)

GO
```

Creación de Llaves foráneas (FK)

Una clave externa o foránea (Foreign Key - FK) de una tabla apunta a una clave primaria de otra tabla. Los Constraints de FOREIGN KEY identifican las relaciones entre las tablas. Las claves externas evitan acciones que podrían dejar filas con valores de claves externas cuando no hay claves candidatas con ese valor (Integridad Referencial)
Ejemplo: A continuación se muestra como se define la FK en la tabla Tb_Factura, donde el campo Cod_cli apunta a la llave primaria de la tabla Tb_Cliente

```
CREATE TABLE [dbo].[Tb_Factura]
(
    [Num_fac] [nvarchar](6) NOT NULL,
    [Fec_fac] [datetime] NOT NULL,
    [Cod_cli] [nvarchar](4) NOT NULL,
    [Fec_can] [datetime] NULL,
    [Est_fac] [nvarchar](50) NOT NULL,
    [Cod_ven] [nchar](3) NOT NULL,
    [Por_Igv] [money] NOT NULL,
    [Fec_Registro] [datetime] NULL,
    [Usu_Registro] [varchar](20) NULL,
    [Fec_Ult_Mod] [datetime] NULL,
    [Usu_Ult_Mod] [varchar](20) NULL,
  PRIMARY KEY
    (
        [Num_fac] ASC
    )
)
GO
```

```
Alter table Tb_Factura
    add  foreign key(Cod_Cli) references Tb_Cliente (Cod_Cli)
go
```

Creación de Constraints

Los constraints (restricciones) son elementos que se definen dentro de las tablas para asegurar la integridad de los datos almacenados. Las mismas PK y FK son constraints dado su carácter restrictivo para las reglas de unicidad (las PK) y de integridad referencial (las FK).
Existen mas tipos de constraints, que son los que aquí se indican :

a) De tipo Check , que permiten establecer reglas para la validar datos almacenados en una columna de la tabla.
b) De tipo Default, que permiten establecer valores por defecto en columnas que permitan valores nulos.
c) De tipo Unique, para establecer que los valores de una columna que no es PK, sea restringida para almacenar valores únicos. Esa columna se convertiría en una Llave Alterna.

Constraints tipo Check

Sintaxis

```
ALTER TABLE
      nombre_tabla
ADD CONSTRAINT      <Nombre constraint>
CHECK
( condición )
```

Ejemplo:
Crear un constraint de tipo check para que el campo de ExParcial de la tabla Tb_Evaluaciones admita solo valores entre 0 y 20:

```
ALTER TABLE Tb_Evaluaciones
    ADD CONSTRAINT chk_ex_parcial CHECK
    (ExParcial  between 0 and 20 )
GO
```

Constraints tipo Default

Sintaxis

ALTER TABLE nombre_tabla
ADD CONSTRAINT <Nombre constraint>
DEFAULT valor_predeterminado
FOR columnaX

Ejemplo:
Crear un constraint de tipo default para el campo Id_Ubigeo de la tabla Tb_Cliente, de tal forma que si se ingrese un nulo por defecto se asuma el ubigeo de Lima (140101)

```
ALTER TABLE Tb_Cliente
        ADD  CONSTRAINT Def_Id_Ubigeo DEFAULT
        '140101' FOR Id_Ubigeo
GO
```

Constraints tipo Unique

Sintaxis

ALTER TABLE nombre_tabla
ADD CONSTRAINT <Nombre constraint>
UNIQUE(columnaX, columnaP, …)

Ejemplo:
Crear un constraint de tipo Unique para que el campo Ruc_Cli de la tabla Tb_Cliente, solo admita valores únicos

```
ALTER TABLE Tb_Cliente ADD CONSTRAINT UQ_Ruc_cli UNIQUE (Ruc_cli)
GO
```

La propiedad Identity

Crea una columna de identidad en una tabla. Esta propiedad se utiliza con las instrucciones CREATE TABLE y ALTER TABLE de Transact-SQL.

EJEMPLO:

A continuación en el ejemplo vamos a crear de la tabla Categoria con el campo IDCategoria de tipo INT y que tenga la propiedad IDENTITY.

```
USE BDEjemplo_01
GO
CREATE TABLE Categoria
(  IDCategoria     INT IDENTITY (1,1) NOT NULL,
   Nombre          VARCHAR(30) NOT NULL
)
GO
```

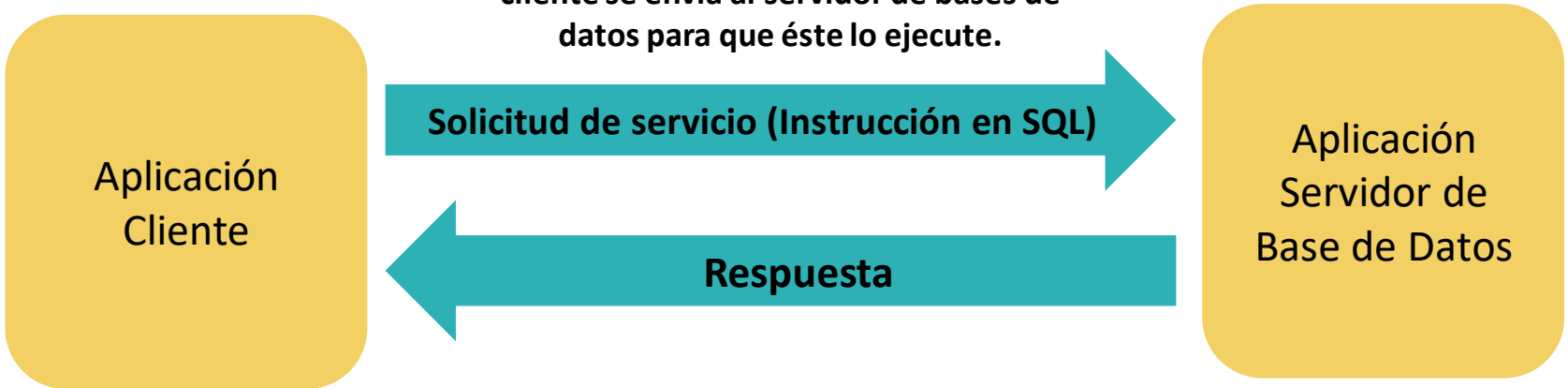
NOTA: El indicador (1,1) establece que se inicia en 1 y se incrementa de 1 en 1. Si por ejemplo se requiere empezar de 10 y que se incremente de 5 en 5 seria (10,5)

LABORATORIO

Realizar junto a su instructor los laboratorios 1 , 2 y 3 de la presente sesión.
Recuerde que para el laboratorio 3 debe ejecutar previamente el script : Lab2- Script_BDVentasTest.sql que es parte del material.

/ RESUMEN Y CONCLUSIONES

El código SQL incrustado en la aplicación cliente se envía al servidor de bases de datos para que éste lo ejecute.



Java
JavaScript
Visual Basic
VBA
C++
C#
Python
…

Oracle DB
MS SQL Server
MS Access
DB2
Informix
Sybase DB
…

/ RESUMEN

RDBMS
Relational DataBase Management System

Resumen

Tipo de Constraint	Descripción
PRIMARY KEY (clave primaria)	Garantiza que cada fila ó registro en una tabla es único(a). La columna ó combinación de columnas definida como clave primaria no permite valores duplicados.
UNIQUE (valor no duplicado)	Garantiza que cada valor en una columna es único. Permite valores únicos
FOREIGN KEY (clave foránea)	Define la columna ó combinación de columnas de una tabla secundaria cuyos valores dependen de la clave primaria de una tabla primaria.
DEFAULT (valor predeterminado)	Establece el valor predeterminado para una columna cuando al insertar una fila no se especifica el valor para dicha columna.
CHECK (regla de validación)	Establece la regla que debe cumplir un valor para que sea un valor aceptable en una columna.

Conclusiones:

- En esta sesión se repaso la creación de tablas en una base de datos, que son los objetos básicos.
- Es importante conocer el lenguaje SQL para poder interactuar con la implementación de los objetos, aunque también podemos recurrir al modo interactivo que nos brinda SQL Server.
- Se hace necesario, para garantizar la integridad de los datos, la aplicación de constraints ya sea de chequeos, default y unique para diversos casos, según sean necesarios.
- Entender el caso de la propiedad Identity para campos autonumerados, lo que nos garantiza que será el propio gestor de base de datos quien asigne los valores para los campos con esta propiedad.