

# PROGRAMACIÓN CON TRANSACT-SQL



CEPSUNI – 2021

LIMA - PERU

## PROGRAMACIÓN CON TRANSACT SQL

Derechos Reservados © 2021 CEPS-UNI

Primera Edición

**LIMA - PERÚ**

## PRESENTACIÓN

TRANSACT-SQL (T-SQL) extiende el estándar de SQL para incluir programación procedimental, funciones de usuario, variables locales, estructuras de control, control de errores, gestión de transacciones, etc.

T-SQL es un lenguaje muy potente que te permite definir casi cualquier tarea que quieras ejecutar sobre la base de datos; incluye características propias de cualquier lenguaje de programación, características que te permiten definir la lógica necesaria para el tratamiento de datos.

En el desarrollo de aplicaciones en general, muchas veces te encuentras con la duda de si la lógica de negocio lo programamos en la aplicación, por ejemplo, con Java, o en la base de datos con procedimientos almacenados. Tal vez aplicar una solución mixta, parte de la lógica en la aplicación y parte en la base de datos.

La posibilidad de que desarrolles las reglas de negocio en procedimientos almacenados puede representar muchas ventajas, por ejemplo, si hay algún cambio en la regla de negocio, puede que sea suficiente la actualización el procedimiento almacenado, y no harías ningún cambio en la aplicación.

Al estudiar este curso te estás preparando como programador para que puedas desempeñarte como programador de base de datos SQL Server.

**Eric Gustavo Coronel Castillo**  
**INSTRUCTOR**

# Índice

<b>BASES DE DATOS .....</b>	<b>9</b>
OBTENER SCRIPTS.....	9
BASE DE DATOS DE RECURSOS HUMANOS – RH .....	9
BASE DE DATOS ACADÉMICA – EDUCA .....	10
BASE DE DATOS ACADÉMICA – EDUTEC .....	10
BASE DE DATOS DE COMERCIAL – NORTHWIND .....	11
BASE DE DATOS EUREKABANK .....	12
<b>CAPÍTULO 1 GESTIÓN DE INDICES.....</b>	<b>13</b>
CONCEPTO .....	13
ACCESO A LOS DATOS .....	14
CRITERIOS PARA CREAR ÍNDICES .....	15
<i>Razones para crear índices.....</i>	<i>15</i>
<i>Razones para no crear índices.....</i>	<i>15</i>
<i>Columnas a indexar .....</i>	<i>16</i>
<i>Columnas que no deben indexarse .....</i>	<i>16</i>
TIPOS DE ÍNDICES .....	17
<i>Índice clustered.....</i>	<i>17</i>
<i>Índice nonclustered.....</i>	<i>17</i>
CREACIÓN DE INDICES .....	18
<i>Creación de índice CLUSTERED.....</i>	<i>18</i>
<i>Creación de índice NONCLUSTERED.....</i>	<i>18</i>
<i>Creación de índice UNIQUE.....</i>	<i>18</i>
MANTENIMIENTO DE INDICES .....	19
<i>Borrar un índice.....</i>	<i>19</i>
<i>Regenerar un índice.....</i>	<i>19</i>
<i>Regenerar los índices de una tabla.....</i>	<i>19</i>
<i>Fragmentación de índices .....</i>	<i>20</i>
<i>Reorganizar índices.....</i>	<i>20</i>
<i>Reorganizar todos los índices de una tabla .....</i>	<i>20</i>
<b>CAPÍTULO 2 FUNDAMENTOS GENERALES .....</b>	<b>21</b>
INTRODUCCIÓN.....	21
<i>T-SQL permite.....</i>	<i>21</i>
<i>T-SQL no permite .....</i>	<i>21</i>
REGLAS DE FORMATO DE LOS IDENTIFICADORES .....	22

LAS EXPRESIONES .....	23
<i>Tipos de operadores</i> .....	23
<i>Resultados de la expresión</i> .....	23
OTROS ELEMENTOS DEL LENGUAJE.....	24
<i>Comentarios</i> .....	24
<i>BEGIN...END</i> .....	24
<b>CAPÍTULO 3 FUNDAMENTOS DE PROGRAMACIÓN .....</b>	<b>25</b>
BLOQUE ANÓNIMO .....	25
FUNCIONES .....	26
<i>Función Escalar</i> .....	26
<i>Función de tabla en línea</i> .....	27
<i>Función de tabla de múltiples instrucciones</i> .....	28
PROCEDIMIENTOS .....	30
ELEMENTOS DE PROGRAMACIÓN .....	32
<i>Variables</i> .....	32
<i>Sentencia de asignación</i> .....	32
EJERCICIOS PROPUESTOS.....	33
<b>CAPÍTULO 4 ESTRUCTURAS DE CONTROL .....</b>	<b>34</b>
BLOQUE.....	34
ESTRUCTURAS CONDICIONALES .....	34
<i>Estructura: IF</i> .....	34
<i>Estructura: CASE</i> .....	35
ESTRUCTURAS DE BUCLE .....	37
<i>Estructura WHILE</i> .....	37
<i>Sentencia BREAK</i> .....	37
<i>Sentencia CONTINUE</i> .....	37
<i>Sentencia GOTO</i> .....	37
<b>CAPÍTULO 5 GESTIÓN DE DATOS .....</b>	<b>39</b>
INSERTANDO DATOS.....	39
<i>Sentencia INSERT</i> .....	39
<i>Insertar una sola fila de datos</i> .....	39
<i>Insertar varias filas de datos</i> .....	40
<i>Insertar datos en una tabla con una columna identidad</i> .....	40
<i>Usar TOP para limitar los datos insertados de la tabla origen</i> .....	41
<i>Ejercicio 1</i> .....	42
<i>Ejercicio 2</i> .....	42
ACTUALIZANDO DATOS .....	43
<i>Sentencia UPDATE</i> .....	43
<i>Usar una instrucción UPDATE simple</i> .....	43
<i>Actualizar varias columnas</i> .....	43
<i>Usar la cláusula WHERE</i> .....	44

<i>Usar la cláusula TOP</i> .....	45
<i>Usar la cláusula WITH common_table_expression</i> .....	47
<i>Especificar una subconsulta en la cláusula SET</i> .....	48
<i>Ejercicio 3</i> .....	49
ELIMINANDO FILAS .....	50
<i>Sentencia DELETE</i> .....	50
<i>DELETE sin la cláusula WHERE</i> .....	50
<i>Usar la cláusula WHERE para eliminar un conjunto de filas</i> .....	50
<i>Usar la cláusula WHERE con una condición compleja</i> .....	51
<i>Utilizar la cláusula TOP para limitar el número de filas eliminadas</i> .....	53
<i>Ejercicio 4</i> .....	55
COMBINANDO DATOS .....	56
SENTENCIA MERGE .....	56
<i>Usar MERGE para realizar operaciones INSERT y UPDATE</i> .....	56
<i>Usar MERGE para realizar operaciones UPDATE y DELETE</i> .....	58
TRANSACCIONES .....	60
<i>Definición</i> .....	60
<i>Propiedades de una Transacción</i> .....	61
<i>Tipos de Transacciones</i> .....	61
<b>CAPÍTULO 6 CONTROL DE ERRORES</b> .....	<b>64</b>
CONTROL DE ERRORES .....	64
<i>Variable: @@ROWCOUNT</i> .....	64
<i>Función: ROWCOUNT_BIG ( )</i> .....	64
<i>Variable: @@ERROR</i> .....	64
<i>Función: RAISERROR ( )</i> .....	65
MANEJO DE EXCEPCIONES .....	66
<i>Estructura TRY/CATCH</i> .....	66
<i>Sentencia: THROW</i> .....	70
REQUERIMIENTOS A RESOLVER .....	71
<i>Requerimiento 1</i> .....	71
<i>Requerimiento 2</i> .....	71
<i>Requerimiento 3</i> .....	72
<b>CAPÍTULO 7 TRABAJANDO CON CURSORES</b> .....	<b>73</b>
TRABAJANDO CON CURSORES .....	73
<i>Declaración</i> .....	73
<i>Abrir un cursor</i> .....	74
<i>Recuperar filas de un cursor</i> .....	74
<i>Cerrar un cursor</i> .....	75
<i>Liberar recursos de un cursor</i> .....	75
CONTROL DE UN CURSOR .....	76
<i>Variable: @@FETCH_STATUS</i> .....	76
<i>Variable: @@CURSOR_ROWS</i> .....	78

<i>Función: CURSOR_STATUS ( )</i> .....	80
BUCLE DE EXTRACCIÓN .....	84
<i>Plantilla</i> .....	84
EJERCICIOS .....	85
<i>Ejercicio 12</i> .....	85
<i>Ejercicio 13</i> .....	85
USO DE TABLAS TEMPORALES.....	86
<i>Variables de tipo tabla</i> .....	86
<i>Tablas temporales locales</i> .....	88
<i>Tablas temporales globales</i> .....	90
EJERCICIOS .....	92
<i>Ejercicio 14</i> .....	92
<i>Ejercicio 15</i> .....	92
<b>CAPÍTULO 8 GESTION DE TRIGGERS .....</b>	<b>93</b>
INTRODUCCIÓN .....	93
TIPOS DE DESENCADENANTES DDL .....	94
<i>Desencadenante Transact-SQL DDL</i> .....	94
<i>Desencadenante CLR DLL</i> .....	94
ÁMBITO DE LOS DESENCADENANTES DDL.....	95
MANTENIMIENTO DE DESENCADENANTES DDL .....	99
<i>Creación de desencadenantes DDL</i> .....	99
<i>Modificar desencadenantes DDL</i> .....	99
DESHABILITAR Y ELIMINAR DESENCADENANTES DDL .....	100
EJEMPLOS.....	101
<i>Ejemplo 32: Log de cambios en el sistema</i> .....	101
<b>CAPÍTULO 9 PRACTICAS CALIFICADAS .....</b>	<b>107</b>
PRACTICA CALIFICADA 1 .....	107
<i>Problema 1</i> .....	107
<i>Problema 2</i> .....	107
<i>Problema 3</i> .....	107
<i>Problema 4</i> .....	107
PRACTICA CALIFICADA 2 .....	108
<i>Problema 5</i> .....	108
<i>Problema 6</i> .....	108
<i>Problema 7</i> .....	108
<i>Problema 8</i> .....	108
PRACTICA CALIFICADA 3 .....	109
<i>Base de datos</i> .....	109
<i>Problema 9</i> .....	109
<i>Problema 10</i> .....	109
<i>Problema 11</i> .....	109
<i>Problema 12</i> .....	109

PRACTICA CALIFICADA 4 .....	110
<i>Problema 13</i> .....	110
<i>Problema 14</i> .....	111
<i>Problema 15</i> .....	112
<b>CURSOS RELACIONADOS .....</b>	<b>113</b>



## BASES DE DATOS

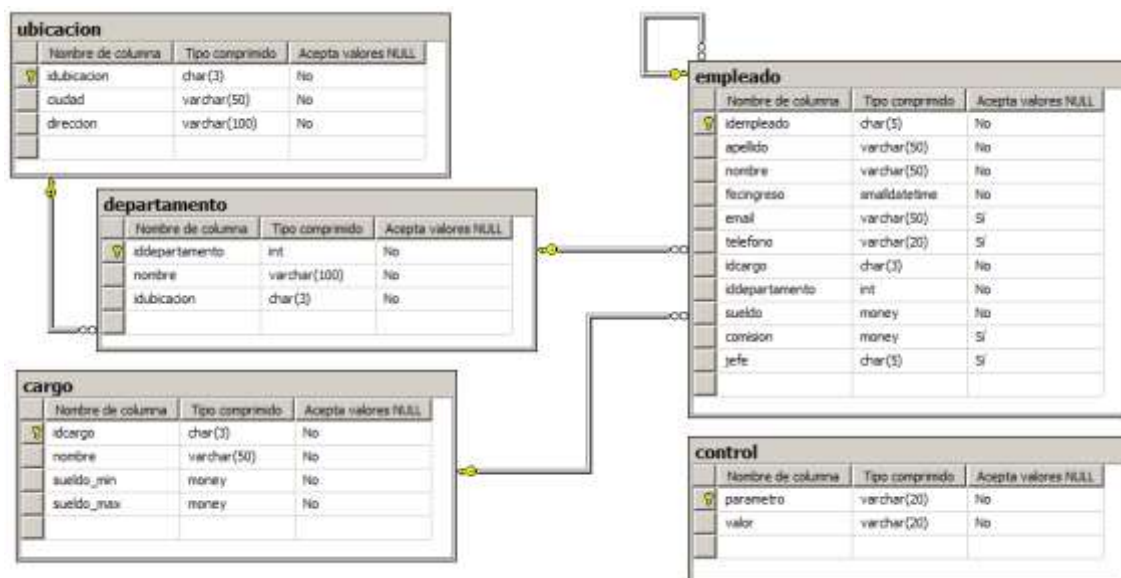
### OBTENER SCRIPTS

Para obtener los scripts de las bases de datos utilizadas en la presente separata utiliza la siguiente URL:

<https://github.com/gcoronelc/databases>

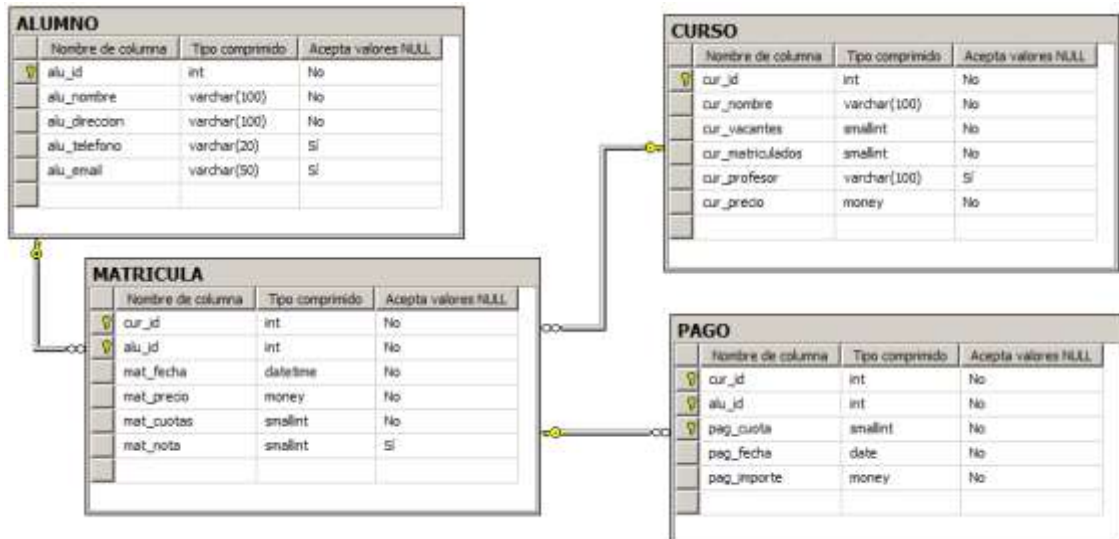
### BASE DE DATOS DE RECURSOS HUMANOS – RH

Base de datos básica de recursos humanos.



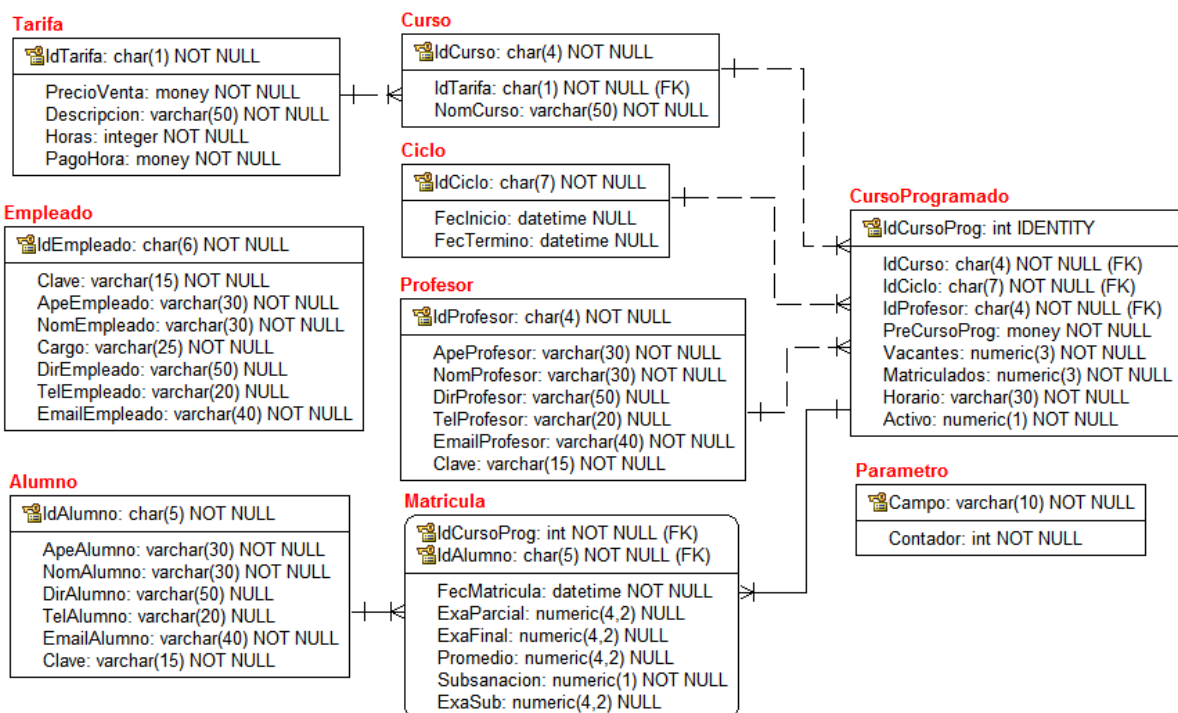
## BASE DE DATOS ACADÉMICA – EDUCA

Base de datos bastante simple de gestión académica.



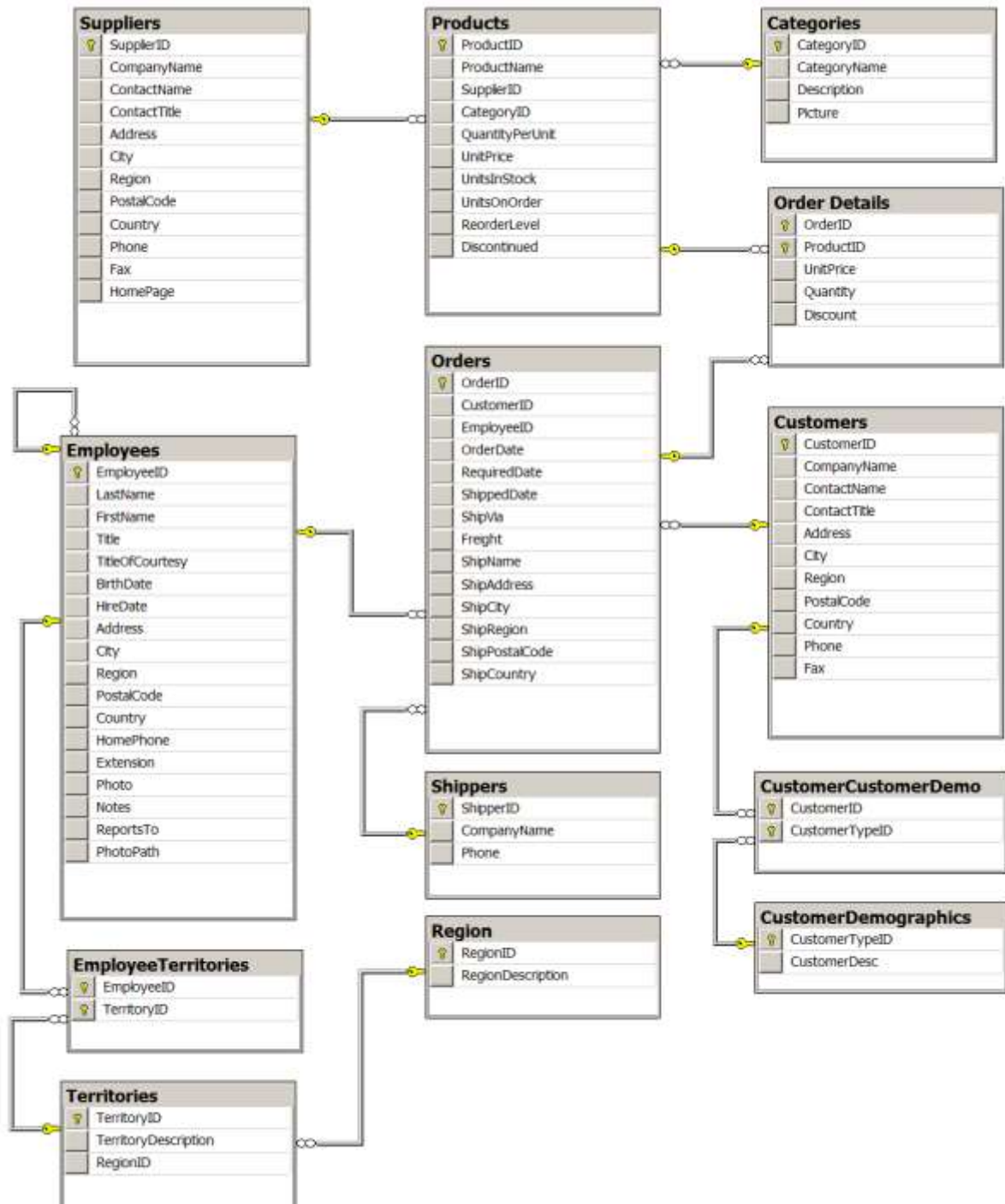
## Base de datos Académica – EDUTEC

Base de datos de gestión de cursos cortos.



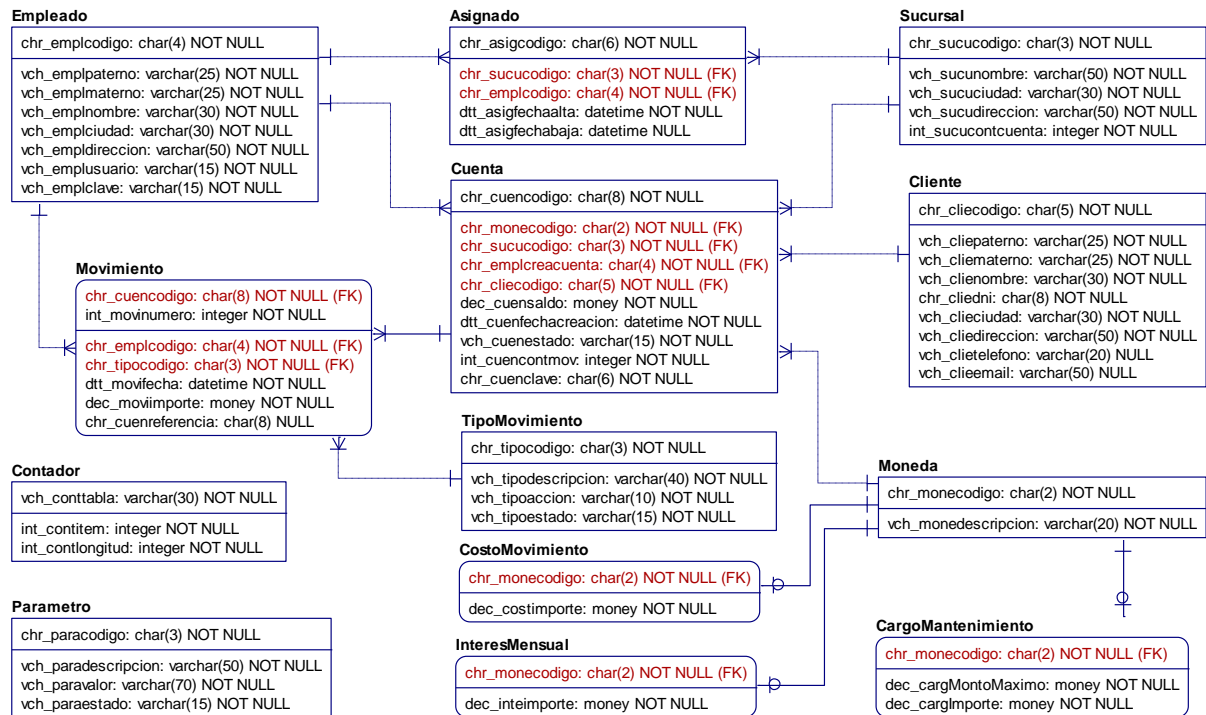
## BASE DE DATOS DE COMERCIAL - NORTHWIND

Base de datos comercial clásica de SQL Server.



## Base de datos EUREKABANK

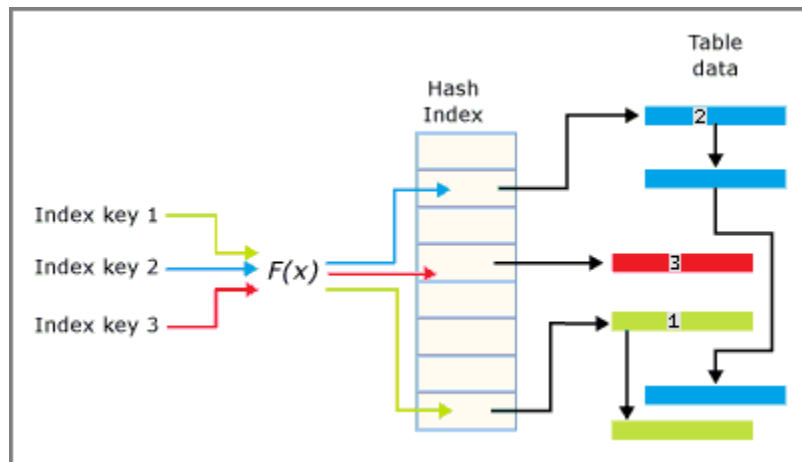
Modelo de base de datos de gestión de cuentas de ahorro de una institución financiera.



## Capítulo 1 GESTIÓN DE INDICES

### CONCEPTO

---



Un índice es una estructura que proporciona acceso rápido a las filas de una tabla en base a los valores de una o más columnas (clave).

Un índice es un conjunto de valores claves y apuntadores lógicos que permite ejecutar búsqueda de registros de modo similar a la manera como buscamos un tema en el índice analítico de un libro. Por lo general, todas las consultas se ejecutan más rápido cuando se utilizan índices.

## ACCESO A LOS DATOS

---

SQL Server accede a los datos de una de estas dos maneras:

- Examinando todas las páginas de datos de la tabla. Este proceso se conoce como **Table Scan**. Cuando SQL Server realiza un Table Scan, esto es lo que sucede:
  - ✓ Lee desde el principio de la tabla.
  - ✓ Examina página a página todas las filas de la tabla.
  - ✓ Extrae las filas que corresponden al criterio de la consulta.
- Usando los índices. Este proceso se conoce como **Index Seek**. Cuando SQL Server usa un índice, esto es lo que sucede:
  - ✓ Atraviesa la estructura de árbol del índice para encontrar las filas que la consulta requiere.
  - ✓ Extrae solo las filas que se corresponden con el criterio de la consulta.

Cuando se envía una consulta, SQL Server determina primero si un índice existe. Entonces, el **Query Optimizer**, el componente responsable de generar el plan de ejecución óptimo para la consulta, determina si es más eficaz examinar la tabla o utilizar el índice para acceder a los datos.

## CRITERIOS PARA CREAR ÍNDICES

---



Si estás considerando crear un índice, aquí tienes algunas recomendaciones.

### Razones para crear índices

Los índices aceleran la recuperación de los datos. Por ejemplo, sin ningún índice, tendrías que recorrer todas las páginas de un libro hasta encontrar la información que estamos buscando.

- Refuerzan la unicidad de las filas.
- Incrementan la velocidad de recuperación de datos:
  - ✓ Los joins se ejecutan más rápido si la columna llave foránea está indexada.
  - ✓ Las consultas ORDER BY y GROUP BY se ejecutan más rápido.

### Razones para no crear índices

En general, cuando ejecutamos operaciones de lectura, los índices favorecen el proceso; cuando las operaciones son de escritura, los índices hacen que el rendimiento del sistema disminuya.

- Consumen espacio de disco.
- Producen sobrecarga en el sistema. Cuando se modifican datos de columnas indexadas, el índice es actualizado automáticamente para reflejar los cambios.

### Columnas a indexar

- Clave primaria.
- Clave foránea.
- Columnas en las que se busca por rango de valores.
- Columnas utilizadas como criterio de ordenamiento.

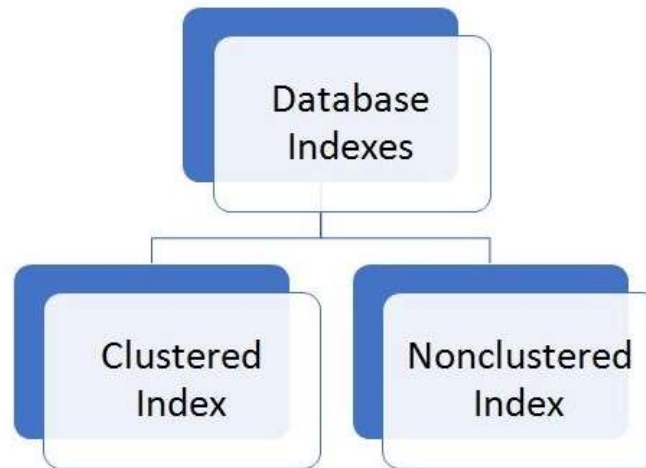
### Columnas que no deben indexarse

- Columnas en las que no se ejecuta búsqueda.
- Columnas con pocos valores únicos o que retornan un gran porcentaje de filas.
- Columnas de tipo bit, text o image.



## TIPOS DE ÍNDICES

---



Hay dos opciones para el almacenamiento físico de sus índices:

- índice agrupado (clustered)
- índice no agrupado (nonclustered)

### Índice clustered

- Ordena físicamente la tabla. Las filas de la tabla se ordenan según el orden de los valores claves del índice clustered.
- Solo se puede definir un índice clustered por cada tabla.

### Índice nonclustered

- Es el tipo de índice por defecto.
- Se reconstruyen automáticamente cuando se crea, se elimina o se redefine el índice clustered.

### Observación:

- Se pueden definir hasta 249 índices por tabla.
- Siempre crear el índice clustered antes que los índices nonclustered.

## CREACIÓN DE INDICES

---

### Sintaxis

```
CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED]  
INDEX NombreDelIndice  
ON NombreDeTabla(Columna1, Columna2, ...)  
[ FILLFACTOR = <factor de relleno> ]
```

Para mayor información consultar:

<https://docs.microsoft.com/en-us/sql/t-sql/statements/create-index-transact-sql>

### Creación de índice CLUSTERED

#### Ejemplo 1

```
CREATE CLUSTERED INDEX EMPLEADO_IDXC  
ON EMPLEADO(ID_PERSONA)
```

### Creación de índice NONCLUSTERED

#### Ejemplo 2

```
CREATE INDEX EMPLEADO_IDXN  
ON EMPLEADO(NOMBRE)
```

### Creación de índice UNIQUE

El índice UNIQUE es aquel que no permite repetición de los valores que conforman el índice.

#### Ejemplo 3

```
CREATE UNIQUE INDEX EMPLEADO_IDXU  
ON EMPLEADO(EMAIL)
```

## MANTENIMIENTO DE INDICES

---



### Borrar un índice

#### Sintaxis

```
DROP INDEX NombreDeTabla.NombreDeIndice
```

### Regenerar un índice

```
ALTER INDEX <index name> ON <table name> REBUILD
```

### Regenerar los índices de una tabla

```
ALTER INDEX ALL ON <table name> REBUILD
```

## Fragmentación de índices

```
SELECT
    c.name "Table name",
    b.name "Index",
    avg_fragmentation_in_percent "Frag (%)",
    page_count "Page count"
FROM sys.dm_db_index_physical_stats (DB_ID(), NULL, NULL, NULL, NULL ) AS a
JOIN sys.indexes AS b ON a.object_id = b.object_id AND a.index_id = b.index_id
JOIN sys.tables c ON b.object_id = c.object_id
ORDER BY 3 DESC;
GO
```

## Reorganizar índices

### Sintaxis

```
ALTER INDEX <index name> ON <table name> REORGANIZE
```

## Reorganizar todos los índices de una tabla

### Sintaxis

```
ALTER INDEX ALL ON <table name> REORGANIZE
```

## CURSOS RELACIONADOS

Para contactarte con algún representante del CEPS-UNI tienes la siguiente URL:

**<https://www.ceps.uni.edu.pe/contacto/>**

A continuación, tienes la lista de cursos relaciones que te pueden interesar:



