



BASE DE DATOS AVANZADO I

Unidad 1: Base de Datos y Tablas

Tema 3: Manejo de Restricciones e Índices





Tema 03: Manejo de Restricciones e Índices

1.3. Tema 3: Manejo de Restricciones e Índices

1.3.1. Constraints o restricciones. Tipos.

1.3.2. Uso del identity

1.3.3. Concepto de índices, tipos de índices

1.3.4. Indices particionados





Capacidades

1. Identifica las opciones de base de datos, tablas e índices.
2. Diseña e implementa una base de datos optimizada, así como las tablas, esquemas e índices garantizando disponibilidad y rendimiento.





1. Constraints o restricciones

¿Qué es *constraint*?

- Es una restricción a un campo de la tabla.
- Logra la integridad de datos.
- Permite implementar las reglas de negocio.
- Toda CONSTRAINT debe tener un nombre definido por el diseñador de la base de datos.





1.1. Tipos de constraints

- Los constraints se clasifican de la siguiente manera:

Integridad de Entidad

Integridad Referencial

Integridad de Datos





1.1. Tipos de constraints

Integridad de Entidad

Constraint PRIMARY KEY

- El valor puede ser el de un campo (simple) o la combinación de varios campos (compuesta)
- Los campos que identifican a PRIMARY KEY deben ser Not Null.
- Restricción:
 - ☐ No permite valores nulos.
 - ☐ No permite valores duplicados.



1.1. Tipos de constraints

Integridad de Entidad

Constraint PRIMARY KEY

- Se define un Primary Key por TABLA
- Se puede definir para una o más COLUMNAS
- Tipo CLUSTERED: La tabla queda ordenada físicamente por su índice.



1.1. Tipos de constraints

Integridad de Entidad

Constraint PRIMARY KEY

TBALUMNO		
codAlum	nomAlum	distAlum
Co1	Juan Peña	Surco
A01	Sara Lopez	Ate
Bo1	Pedro Urbina	Comas

PRIMARY KEY CLUSTERED

TBALUMNO		
codAlum	nomAlum	distAlum
A01	Sara Lopez	Ate
Bo1	Pedro Urbina	Comas
Co1	Juan Peña	Surco

PRIMARY KEY NONCLUSTERED

TBALUMNO		
codAlum	nomAlum	distAlum
Co1	Juan Peña	Surco
A01	Sara Lopez	Ate
Bo1	Pedro Urbina	Comas



1.1. Tipos de constraints

Integridad de Entidad

Creación Constraint PRIMARY KEY

- Existen 02 formas:
 - Al momento de crear la tabla (CREATE TABLE).
 - Crear sobre una tabla existente (ALTER TABLE).



1.1. Tipos de constraints

Integridad de Entidad

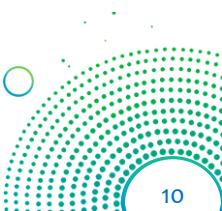
Creación Constraint PRIMARY KEY

Al momento de crear la tabla (CREATE TABLE).

En la base de datos BDTravel crear tabla país en el esquema MKT y define al campo idPaís como Primary Key Clustered.

```
Use BDTravel
go

Create Table MKT.Pais
(
  idPaís int not null,
  nompais varchar(50) not null,
  constraint PKPais Primary Key (idPaís)
)
go
```





1.1. Tipos de constraints

Integridad de Entidad

Creación Constraint PRIMARY KEY

- Crear sobre una tabla existente (ALTER TABLE).

En la base de datos BDTravel crear tabla Tienda . Luego altera la tabla, define al campo idTienda como Primary Key NonClustered.

1

```
Use BDTravel
go

Create Table MKT.Tienda
(
    idTienda int not null,
    nomTienda varchar(50) not null,
    dirTienda varchar(60) not null
)
go
```

2

```
Use BDTravel
go

Alter Table MKT.Tienda
Add Constraint PKTienda
Primary Key Nonclustered(idTienda)
go
```

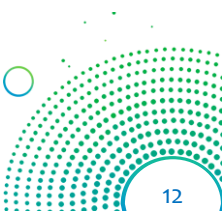


1.1. Tipos de constraints

Integridad Referencial

Constraint FOREIGN KEY

- Un Foreign Key permite definir una clave externa, que es una columna o combinación de columnas, que se utiliza para establecer y exigir un vínculo entre los datos de dos tablas.
- Evidencia una relación física entre dos entidades
- En una relación “Uno a Muchos” la llave del lado “Uno” es referenciada desde el lado “Muchos”



1.1. Tipos de constraints

Constraint FOREIGN KEY

Integridad Referencial

TBPAIS	
idPais	nomPais
10	PERU
20	ARGENTINA
30	COLOMBIA

REFERENCE

TBCLIENTE		
idCliente	nomCliente	idPais
Co1	Sara Lopez	10
Co2	Pedro Urbina	20
Co3	Juan Peña	10

FOREIGN KEY

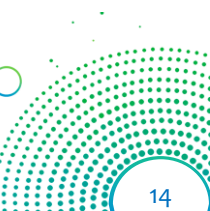


1.1. Tipos de constraints

Integridad Referencial

Creación Constraint FOREIGN KEY

- Existen 02 formas:
 - Al momento de crear la tabla (CREATE TABLE).
 - Crear sobre una tabla existente (ALTER TABLE).





1.1. Tipos de constraints

Creación Constraint FOREIGN KEY

Integridad Referencial

- Al momento de crear la tabla (CREATE TABLE).

Crear la tabla Cliente, definiendo el campo idPais como FOREIGN KEY, relacionando con la tabla Pais.

```
-- Create Table MKT.Cliente
(
  idCliente    char(5) not null,
  nomCliente   Varchar(50) not null,
  idPais       int null,
  Constraint FKPaísCliente Foreign Key (idPais)
               references MKT.Pais
)
go
```





1.1. Tipos de constraints

Integridad Referencial

Creación Constraint FOREIGN KEY

- Al momento de crear la tabla (CREATE TABLE).

Crear la tabla Proveedor, definiendo el campo idPais como FOREIGN KEY, incluya la opción de eliminación en cascada.

```
Create Table MKT.Proveedor
(
  idProveedor      char(5) not null,
  nomProveedor     varchar(40) not null,
  idPais           int null,
  Constraint FKPaísProveedor Foreign Key (idPais)
                  References MKT.País
                  On Delete Cascade
)
go
```

ON DELETE CASCADE
Permitirá que si se elimina un registro de la tabla país, se eliminará también los registros dependientes de la tabla Proveedor



1.1. Tipos de constraints

Integridad Referencial

Creación Constraint FOREIGN KEY

- Crear sobre una tabla existente (ALTER TABLE).

Crear la tabla Empleado, luego alterar la tabla para definir al campo idTienda como FOREIGN KEY, incluya la opción de actualización en cascada.

```
CREATE TABLE MKT.Empleado
(
  codEmple      char(5) not null,
  nomEmple      varchar(50) not null,
  idTienda      int null
)
go
```



1.1. Tipos de constraints

Integridad Referencial

Creación Constraint FOREIGN KEY

- Crear sobre una tabla existente (ALTER TABLE).

```
Alter Table MKT.Empleado  
add Constraint FKTiendaEmpleado  
Foreign Key (idTienda) References MKT.Tienda  
On Update Cascade  
go
```

ON UPDATE CASCADE

Permitirá que si se actualiza el identificador de la tabla país, también se actualizará los idpais de los registros dependientes de la tabla Proveedor



1.1. Tipos de constraints

Integridad de Datos

Constraint UNIQUE

- Garantiza que no se escriban valores duplicados en columnas específicas que no forman parte de una clave principal.
- En una tabla se pueden definir varias restricciones UNIQUE.
- UNIQUE admiten valores NULL. Sin embargo, de la misma forma que cualquier valor incluido en una restricción UNIQUE, sólo se admite un valor NULL por columna.

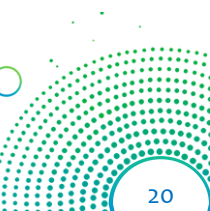


1.1. Tipos de constraints

Integridad de Datos

Creación Constraint UNIQUE

- Existen 02 formas:
 - Al momento de crear la tabla (CREATE TABLE).
 - Crear sobre una tabla existente (ALTER TABLE).





1.1. Tipos de constraints

Integridad de Datos

Creación Constraint UNIQUE

❓ Al momento de crear la tabla (CREATE TABLE).

Crear la tabla conductor, asegurando que el campo nro. de breveté (brvCond) sea único.

```
Create Table MKT.TBConductor
(
  codCond char(5) not null,
  nomCond varchar(50) not null,
  apeCond varchar(50) not null,
  brvCond char(10) not null,
  Constraint UQBrevete Unique (brvCond)
)
go
```

1.1. Tipos de constraints

Integridad de Datos

Creación Constraint UNIQUE

- Existen 02 formas:

Crear sobre una tabla existente (ALTER TABLE).

Alterar la tabla Conductor para asegurar que el nombre y apellido sean únicos (compuesto).

```
Alter Table MKT.TBConductor  
add Constraint UQNomAPConductor  
Unique (nomCond, apeCond)  
go
```



1.1. Tipos de constraints

Integridad de Datos

Constraint DEFAULT

- Se asigna un valor predeterminado a la columna.
- En caso de dejarse vacía al momento de la creación de la fila se registra dicho valor.



1.1. Tipos de constraints

Integridad de Datos

Creación Constraint DEFAULT

- Existen 02 formas:
 - ❑ Al momento de crear la tabla (CREATE TABLE).
 - ❑ Crear sobre una tabla existente (ALTER TABLE).



1.1. Tipos de constraints

Integridad de Datos

Creación Constraint DEFAULT

Al momento de crear la tabla (CREATE TABLE).

Crear la tabla Postulante, definiendo como valor predeterminado 'Soltero' al campo estado civil.

```
CREATE TABLE MKT.Postulante
(
  codPos char(5) not null,
  nomPos varchar(50) not null,
  eciPos varchar(30) Constraint DFEcivilPos Default 'Soltero',
  frePos Date
)
go
```



1.1. Tipos de constraints

Crear sobre una tabla existente (ALTER TABLE).

Creación Constraint DEFAULT

Integridad de Datos

- Alterar la tabla postulante para definir como valor predeterminado la fecha del sistema al campo fecha de registro (frePos).

```
Alter Table MKT.Postulante  
add Constraint DFFecRegistroPos  
Default Getdate() For frePos  
go
```

1.1. Tipos de constraints

Integridad de Datos

Constraint CHECK

- CHECK exige la integridad del dominio mediante la limitación de los valores que puede aceptar una columna a partir de una expresión lógica que no se basa en datos de otra columna.
- Este rango puede ser un conjunto de valores estáticos : ('MASCULINO','FEMENINO')
- Este rango puede ser una condición de validez : =,<,>,>=,<=.
- Este rango puede tener relación con otras entidades.

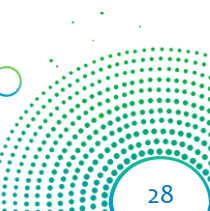


1.1. Tipos de constraints

Integridad de Datos

Creación Constraint CHECK

- Existen 02 formas:
 - Al momento de crear la tabla (CREATE TABLE).
 - Crear sobre una tabla existente (ALTER TABLE).





1.1. Tipos de constraints

Integridad de Datos

Creación Constraint CHECK

- Al momento de crear la tabla (CREATE TABLE).

Crear tabla Producto con restricción al código para admitir 5 dígitos siendo el primero la letra C y los otros 4 numéricos.

```
Create Table MKT.Producto
(
codPro  char(5) not null,
nomPro  varchar(50) not null,
fepPro  date    not null,
fevPro  date    not null,
prePro  smallmoney not null,
Constraint CKcodProducto Check (codPro Like 'P[0-9][0-9][0-9][0-9]')
)
go
```



1.1. Tipos de constraints

- Crear sobre una tabla existente (ALTER TABLE).

Creación Constraint CHECK

Integridad de Datos

- Alterar la tabla Producto para crear restricción al campo fecha de vencimiento debe ser mayor al campo fecha de producción y también el precio debe ser mayor a 0.

```
Alter Table MKT.Producto
add Constraint CKfecVencimientoPro
Check (fevPro > fepPro),
      Constraint CKpreProducto
Check (prePro > 0)
go
```



1.1. Tipos de constraints

Crear sobre una tabla existente (ALTER TABLE).

Creación Constraint CHECK

Integridad de Datos

- Alterar la tabla Producto para crear restricción al campo fecha de producción, deberá ser menor o igual a fecha actual. No aplicar comprobación de regla a datos existentes.

```
Alter Table MKT.Producto with NoCheck  
add Constraint CKfecProduccionPro  
Check (fepPro >= getdate())  
go
```

1.1. Tipos de constraints

Deshabilitar / Habilitar CHECK

Integridad de Datos

```
--Deshabilita Check
[- Alter Table MKT.Producto
  | noCheck Constraint CKcodProducto
  | go

--Habilita Check
[- Alter Table MKT.Producto
  | Check Constraint CKcodProducto
  | go
```


1.1. Tipos de constraints

Visualizar las Restricciones creadas.

```
- Select * from sys.check_constraints  
  Select * from sys.key_constraints  
  Select * from sys.default_constraints  
go
```



1.1. Tipos de constraints

Eliminar Restricciones creadas. ALTER TABLE/DROP CONSTRAINT

- Sintaxis:

`ALTER TABLE nombreTabla`

`DROP CONSTRAINT nombreConstraint`

`GO`

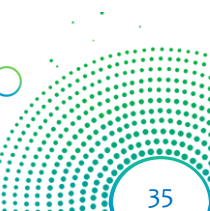
- Ejemplo: Eliminar restricción.

```
Alter Table MKT.Producto  
Drop Constraint CKcodProducto  
go
```



2. Uso de identity

- *Identity* (valor correlativo)
- Columna se convierte en un correlativo automático.
- Debe ser de tipo numérico entero.
- Es una columna de solo lectura.
- Una entidad puede tener solo una columna Identidad.
- En algunos casos puede coincidir con el PRIMARY KEY.





2. Uso de identity

- Debe tener un Valor Inicial y un Incremento

Tabla: Ventas.PedidosCabe

idPedido	FechaPedido	MontoPedido
5	01/08/2000	8900,00
10	02/08/2000	780,00
15	02/08/2000	450,00

IDENTIDAD 5 : 5 (Inicio e Incremento)



2. Uso de identity

Crear campo IDENTITY (valor correlativo).

- Crear tabla Ticket con campo identity
- Debe tener un Valor Inicial 1000 y un Incremento de 1

```
CREATE TABLE MKT.Ticket
(
    nroTicket      int Identity (1000, 1),
    fexTicket       datetime,
    fevTicket       datetime,
    nomEvento       varchar(50),
    valTicket       smallmoney
)
go
```



2. Uso de identity

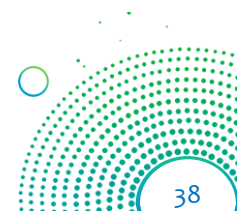
Resetear IDENTITY .

- Para resetear el identity de la tabla, ejecuta la función CheckIdent del comando DBCC.

DBCC CheckIdent (*nombreTabla*, RESEED, *nuevoValor*)

- En el siguiente ejemplo, resetea el valor de identidad a 2000 de la tabla Ticket, campo número de ticket:

```
DBCC CheckIdent ('MKT.Ticket', Reseed, 2000)  
go
```



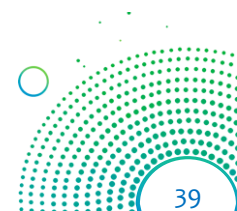


2. Uso de identity

Visualizar campos tipo IDENTITY

- Se utiliza catálogo sys.Identity_Columns

```
Select * from sys.identity_columns  
go
```



2. Uso de identity

Desactivar / Activar IDENTITY.

```
--Desactiva Identity
Set Identity_Insert BDTravel.MKT.Ticket on
go

--Inserta registro incluso campo IDENTITY
Insert MKT.Ticket
(nroTicket,fexTicket,fevTicket,nomEvento,valTicket)
Values
(6666,getdate(),getdate()+1,'Conc Phill Coll','200')
go

--Activa Identity
Set Identity_Insert BDTravel.MKT.Ticket off
go
```




3. Gestión de índice

- ¿Qué es un índice ?
- Un índice es una estructura de datos que permite acceder a diferentes filas de una misma tabla a través de un campo (o campos clave).
- Un índice permite un acceso mucho más rápido a los datos.
- Cada tabla se divide en páginas de datos, o bien en páginas a las que podemos acceder rápidamente a través de un índice.
- Está idea es la que se aplica en el mundo de las bases de datos, la información esta guardada en una tabla (el libro) que tiene muchas hojas de datos (las páginas del libro), con un índice en el que podemos buscar la información que nos interesa.



3. Gestión de índice

- *¿Cómo se trabajan los índices?*
- **Escenario:** Si queremos buscar la palabra zapato, ¿qué hacemos? . Tenemos 02 opciones:
 - Opción 1: Leemos todo el diccionario hasta encontrar la palabra.
 - Opción 2: Buscamos en el índice en que página está la letra z, luego buscamos esa página.
- **Solución:** Ni dudar, la opción dos es la correcta, y es de este modo como se utiliza un índice en las bases de datos



3. Gestión de índice

- *Cómo se trabajan los índices?*
- Los índices se actualizan automáticamente cuando realizamos operaciones de escritura en la base de datos. Este es un aspecto muy importante de cara al rendimiento de las operaciones de escritura, ya que además de escribir los datos en la tabla se escribirán también en el índice.
- Un número elevado de índices hará más lentas estas operaciones. Sin embargo, salvo casos excepcionales, el beneficio que aportan los índices compensa (de largo) esta penalización.





3. Gestión de índice

Creación de ÍNDICES

- Sintaxis:

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ]  
INDEX index_name    ON {tabla | vista} ( column [ ASC | DESC ] [ ,...n ] )  
[ INCLUDE ( column_name [ ,...n ] ) ]  
[ WITH  
    [ PAD_INDEX = { ON | OFF } ]  
    [ FILLFACTOR = fillfactor ]  
    [ ALLOW_ROW_LOCKS = { ON | OFF } ]  
    [ ALLOW_PAGE_LOCKS = { ON | OFF } ]  
    [ ON scheme_partition(columna) | FILEGROUP ]
```



3. Gestión de índice

Creación de ÍNDICES .

- Donde:

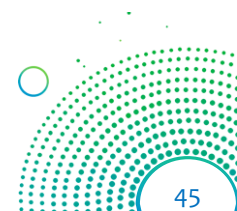
INCLUDE: Especifica las columnas que no son claves que se añaden en el nivel hoja del índice no agrupado.

PAD_INDEX: especifica el relleno del índice, el valor predeterminado es OFF

FILLFACTOR: especifica un porcentaje que indica que porcentaje de la hoja debe estar completa, el valor se encuentre entre 1 al 100, por defecto es 0.

ALLOW_ROW_LOCKS: especifica si los bloqueos de fila se permite, el valor por defecto es ON.

ALLOW_PAGE_LOCKS: especifica si los bloqueos de página están permitidos, por defecto es ON:





3. Gestión de índice

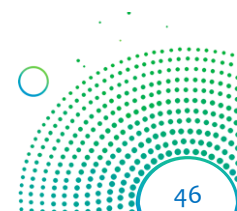
Creación de ÍNDICES

- Ejemplos
- Crear un índice al campo nombre de evento de la tabla Ticket.

```
= Create Index IDXNomEventoTicket  
| On MKT.Ticket (nomEvento)  
go
```

- Crear un índice único al campo nombre de producto de la tabla Producto.

```
= Create Unique Index IDXNomProducto  
| On MKT.Producto (nomPro)  
go
```



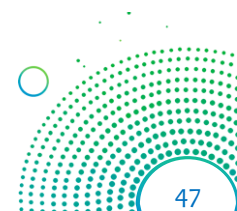


3. Gestión de índice

Creación de ÍNDICES

- *Ejemplos*
- Crear un índice al campo nombre de postulante incluyendo los campos alternativos estado civil y fecha de registro.

```
Create Index IDXNomPostulante  
On MKT.Postulante(nomPos)  
Include (eciPos, frePos)  
go
```





3. Gestión de índice

Modificación de ÍNDICES.

- *Ejemplos*
- Modificar el índice IDXNomEventoTicket de la tabla Ticket para reconstruir su estructura y reactivarla.

```
Alter Index IDXNomEventoTicket  
On MKT.Ticket Rebuild  
go
```





3. Gestión de índice

Modificación de ÍNDICES

- *Ejemplos*
- Desactivar el índice IDXNomEventoTicket de la tabla Ticket.

```
Alter Index IDXNomEventoTicket  
On MKT.Ticket Disable  
go
```





3. Gestión de índice

Modificación de ÍNDICES

- Ejemplos
- Desactivar todos los índices de la tabla Ticket.

```
Alter Index All On MKT.Ticket  
Disable  
go
```





3. Gestión de índice

Visualizar los ÍNDICES. Ejemplos

Listar todo los índices creados en la Base de datos.

```
Select  name          [Índice],  
        type_desc     [Tipo]  
from sys.indexes  
where   name like 'IDX%' or  
        name like 'UQ%' or  
        name like 'PK%'  
go
```





Ejercicio 1

- En la empresa Recobra, se decidió implementar restricciones a la tabla Empleado que se creará en el schema RRHH.
 - Llave primaria NonClustered.
 - Nombre y apellidos debe ser únicos.
 - Sueldo no debe ser menor de 1000, ni tampoco exceder los 15000.
 - Valor predeterminado fecha de ingreso con fecha actual del sistema.



Ejercicio 1: Solución

```
Use BDBRecobra  
go
```

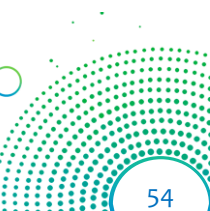
```
(create Table RRHH.Empleado  
(IdEmp      char (8)  not null,  
 nomEmp     varchar (50) not null,  
 apeEmp     varchar (50) not null,  
 fnaEmp     date not null,  
 finEmp     date constraint DFingresoEmple Default getdate (),  
 sueEmp     smallmoney,  
  
 constraint PKEmpleado Primary Key NonClustered (IdEmp),  
 constraint UQnomApEmple Unique (nomEmp, apeEmp),  
 constraint CKsueludoEmp Check (sueEmp between 1000 and 15000)  
  
go
```





Conclusiones

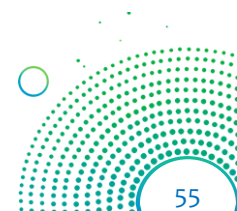
- Las restricciones son muy importante porque nos permiten implementar las reglas de negocio, así también aseguramos calidad de datos.
- Los índices ayudan a obtener mejor rendimiento de respuestas en consultas solicitadas, saber utilizar esta importante estructura y no crear excesivamente porque atenta el tema de almacenamiento.





Bibliografía

- Microsoft (2017) Constraint. Recuperado de: <https://docs.microsoft.com/en-us/sql/relational-databases/tables/disable-check-constraints-with-insert-and-update-statements?view=sql-server-2017>
- Microsoft (2017) Clustered and nonclustered indexes. Recuperado de: <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-2017>



GRACIAS

**SEDE LIMA CENTRO**

Av. Uruguay 514
Cercado – Lima
Teléfono: 419-2900

SEDE INDEPENDENCIA

Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA

Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO

Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO

Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE BELLAVISTA

Av. Mariscal Oscar R. Benavides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA

Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535