Queries auxiliares

```
SELECT
    object_name(cols.object_id) tabla
    ,cols.name columna
    ,ind.name indice
    ,ind.type_desc tipo
    ,ind.is_unique
    FROM
    sys.columns cols, sys.indexes ind , sys.index_columns ind_cols
    where
    cols.object_id = ind.object_id
    and cols.object_id = ind_cols.object_id
    and cols.column_id = ind_cols.column_id
    and ind.index_id = ind_cols.index_id
    and object_name(cols.object_id) LIKE 'Employee'
    order by object_name(cols.object_id), ind.name;
```

```
SELECT TABLE_NAME, COLUMN_NAME, IS_NULLABLE, DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'WorkOrder';
```

“If a column is unique, then it will have the highest possible selectivity, and the selectivity degrades as the level of uniqueness decreases. A column such as ‘gender’, for example, will likely have a low selectivity”.

Convenciones de nomenclatura de índice de SQL Server

- PK_ Primary Key, clustered
- AK_ unclestered, unique
- IX_ unclestered, no unique

Ctrl + M ← hace la consulta y dice el plan que ejecutó

O

View > Command Palette > Run current query with actual plan

--consulta 1

```
select NationalIDNumber, BusinessEntityID from HumanResources.Employee
where NationalIDNumber= '121491555';

select NationalIDNumber, BusinessEntityID from HumanResources.Employee
where NationalIDNumber= 121491555;
```

```
En la primera se busca por un string en la segunda por un entero. Pero
NationalIdNumber es un varchar. Como contraejemplo, si el string que se guarda es el
'000123', no voy a ganar mucho convirtiendo el número 123 a string.
```

--consulta 2

```
select count(UnitPrice) from sales.SalesOrderDetail;

select count(CarrierTrackingNumber) from sales.SalesOrderDetail;

En uno la columna es nulleable (CarrierTrackingNumber) y en el otro no (UnitPrice).
Para la que no es nulleable el count me va a decir cuántos registros tiene la tabla.
Usa el índice unclustered que es más chico.
```

--consulta 3

```
select p.ProductNumber from Sales.SpecialOffer so
join Sales.SpecialOfferProduct sop on so.SpecialOfferID = sop.SpecialOfferID
join Production.Product p on sop.ProductID = p.ProductID

select * from Sales.SpecialOffer so
join Sales.SpecialOfferProduct sop on so.SpecialOfferID = sop.SpecialOfferID
join Production.Product p on sop.ProductID = p.ProductID

En la primera consulta estamos buscando una sola columna del join y en la segunda
todas. INTEGRIDAD REFERENCIAL: el motor tiene los datos de cómo están vinculados los
datos entre sí. Ver que ProductNumber es único.
La primera devuelve los ProductNumber que surgen del matcheo entre SpecialOfferProduct
y Product, a través de ProductID. Pero SpecialOffer no tiene vínculo con Product. Por
lo tanto, no agrega nada que haga ese join.
Tener en cuenta: SpecialOfferID es la clave de SpecialOffer. Por lo tanto, los
SpecialOfferID que tenga la tabla SpecialOfferProduct seguro los va a encontrar en
SpecialOffer. Todo lo que está en SpecialOfferProduct también está en SpecialOffer.
En SpecialOfferProduct hay dos claves foráneas: una va hacia SpeccialOffer y la otra
va hacia Product.
```

--consulta 4

```
select AddressID, City, StateProvinceID, ModifiedDate
from  Person.Address
where StateProvinceID = 32

select AddressID, City, StateProvinceID, ModifiedDate
from Person.Address
where StateProvinceID = 20

Cambia la cantidad de registros a devolver. Hay pocas personas en la provincia 32.
Puedo usar el índice.
```