

Taller	SQL Server Implementación
Docente	Ing. Eric Gustavo Coronel Castillo
Tema	Casos Especiales

## CREACIÓN DE TABLAS A PARTIR DE CONSULTAS

### Sintaxis

```
SELECT < lista_de_columnas >
INTO < nueva_tabla >
FROM . . .
```

### Ejercicios

1. Crear una tabla que contenga la siguiente información:
  - Código de departamento
  - Nombre del departamento
  - Importe de planilla
  - Importe de planilla proyectada con un aumento de 15%

Base de datos a utilizar: RH

Nombre de tabla: RH.DBO.PLANILLA

La consulta a la tabla PLANILLA debe dar un resultado similar al siguiente:

```
SELECT * FROM RH.DBO.PLANILLA;
```

	codido	nombre	planilla actual	planilla proyectada
1	100	Gerencia	33000,00	37950,00
2	101	Contabilidad	25000,00	28750,00
3	102	Investigacion	41800,00	48070,00
4	103	Ventas	37000,00	42550,00
5	105	Sistemas	26500,00	30475,00

2. Crear una tabla con el resumen de los pedidos por ciudad, durante cada año. Las columnas que debe tener la nueva tabla son:
  - Año
  - Ciudad

- Cantidad de pedidos
- Cantidad de artículos
- Importe total

## OPERADORES DE CONJUNTOS

### Operador: UNION

#### Sintaxis

```
< consulta_1 >  
UNION [ALL]  
< consulta_2 >  
UNION [ALL]  
< consulta_3 >  
.  
.  
.
```

#### Ejercicio

3. En una sola consulta combine los datos de los alumnos y profesores de la base de datos EDUCA. La información a mostrar es:
  - Categoría: Puede ser Alumno o Profesor
  - Nombre: debe ser el nombre del alumno o profesor

	CATEGORIA	NOMBRE
1	Alumno	GLADYS REYES CORTIJO
2	Alumno	JHON VELASQUEZ DEL CASTILLO
3	Alumno	OSCAR ALVARADO FERNANDEZ
4	Alumno	SARA RIEGA FRIAS
5	Alumno	YESENIA VIRHUEZ
6	Profesor	Gustavo coronel
7	Profesor	Sergio Matsukawa

## Operador: EXCEPT

### Sintaxis

```
< Consulta_1 >  
EXCEPT  
< Consulta_2 >
```

### Ejercicio

4. Se necesita un listado de los alumnos (solo código) que no se han matriculado en el curso **SQL Server Implementación**. Utilice la base de datos EDUCA.

	alu_id
1	1
2	2

## Operador: INTERSECT

### Sintaxis

```
< Consulta_1 >  
INTERSECT  
< Consulta_2 >
```

5. Se necesita un listado de los alumnos (solo código) que están matriculados en los cursos **SQL Server Implementación** y **SQL Server Administración**. Utilice la base de datos EDUCA.

	alu_id
1	3
2	4
3	5

## CLAUSULA: TOP

### Sintaxis

```
TOP (expression) [PERCENT] [ WITH TIES ]
```

### Ejercicios

6. Desarrolle una consulta para averiguar quiénes son los trabajadores que tienen el sueldo más bajo. Base de datos RH.
7. Desarrolle una consulta para averiguar quiénes son los alumnos con la menor nota. Base de datos EDUCA.

## CLAUSULA: DISTINCT

### Sintaxis

```
[ ALL | DISTINCT ]
```

8. Desarrolle una consulta que muestre el listado de los profesores. Base de datos EDUCA.

## FUNCIÓN: CASE

### Sintaxis

Simple CASE expression:

```
CASE input_expression  
  WHEN when_expression THEN result_expression [ ...n ]  
  [ ELSE else_result_expression ]  
END
```

Searched CASE expression:

```
CASE
  WHEN Boolean_expression THEN result_expression [ ...n ]
  [ ELSE else_result_expression ]
END
```

### Ejercicios

9. Realice una consulta que califique la nota de cada alumno según el siguiente cuadro:

Nota	Calificación
[ 0, 10>	Malo
[10, 14>	Regular
[14, 18>	Bueno
[18,20]	Excelente

Base de datos EDUCA.

10. Se necesita saber la cantidad de alumnos matriculados, aprobados y desaprobados por cada trimestre del año 2012, debe tomar en cuenta la fecha de matrícula.

El formato del resultado es:

TRIMESTRE	MATRICULADOS	APROBADOS	DESAPROBADOS
1			
2			
3			
4			

## OPERADOR: PIVOT

### Sintaxis

```
SELECT <columna no dinamizada>,  
    [primera columna dinamizada] AS <nombre de columna>,  
    [segunda columna dinamizada] AS <nombre de columna>  
    . . .  
    [última columna dinamizada] AS <nombre de columna>  
FROM  
    (<la consulta SELECT que genera los datos>)  
    AS <alias de la consulta de origen>  
PIVOT  
(  
    <función de agregación>(<columna que se agrega>)  
    FOR  
    [ < columna que contiene los valores que se convertirán  
        en encabezados de columna> ]  
    IN ([primera columna dinamizada], [segunda columna dinamizada]  
        ... [última columna dinamizada])  
) AS <alias de la tabla dinamizada>  
<cláusula ORDER BY opcional>;
```

### Ejercicios

11. Encontrar el ingreso por mes de cada curso. Base de datos EDUCA.
12. Encontrar el ingreso por trimestre de cada año. Base de datos EDUCA.
13. Encontrar la cantidad de empleados que han ingresado por trimestre en cada año. Base de datos RH.

## COMMON TABLE EXPRESSION (CTE)

### Sintaxis

```
[ WITH <common_table_expression> [ ,...n ] ]  
<common_table_expression>::=  
expression_name [ ( column_name [ ,...n ] ) ]  
AS  
( CTE_query_definition )
```

### Ejercicios

14. Encontrar el empleado que tiene el menor salario por departamento. Base de datos RH.
15. Encontrar el alumno con la mejor nota por curso. Base de datos EDUCA.