

1) Utiliza las tablas *HumanResources.Employee* y *Person.Person* para crear una consulta que obtenga los nombres de los empleados (*FirstName* y *LastName*) y el título de su puesto (*JobTitle*) usando un INNER JOIN.

```
select p.FirstName,
       p.LastName,
       hre.JobTitle
from HumanResources.Employee hre
     INNER JOIN Person.Person p
       ON hre.BusinessEntityID = p.BusinessEntityID;
```

100 %

Results Messages

	FirstName	LastName	JobTitle
1	John	Kane	Production Technician - WC10
2	Christopher	Hill	Production Technician - WC10
3	Jinghao	Liu	Production Supervisor - WC50
4	Alice	Ciccu	Production Technician - WC50
5	Jun	Cao	Production Technician - WC50
6	Suroor	Fatima	Production Technician - WC50
7	John	Evans	Production Technician - WC50
8	Shelley	Dyck	Production Technician - WC20
9	Terrence	Earls	Production Technician - WC20
10	Jeff	Hay	Production Supervisor - WC45
11	Kirk	Koenigsbauer	Production Technician - WC45

Query executed successfully.

2) Utiliza las tablas *Sales.Customer* y *Person.Person* para crear una consulta que obtenga los nombres de los clientes (*FirstName* y *LastName*). Asegúrate de incluir todos los clientes, incluso aquellos que no tienen un registro en la tabla *Person.Person* usando un LEFT JOIN.

```
SELECT sc.CustomerID,
       p.FirstName,
       p.LastName
FROM Sales.Customer sc
     LEFT OUTER JOIN Person.Person p
       ON p.BusinessEntityID = sc.PersonID
```

100 %

Results Messages

	CustomerID	FirstName	LastName
1	29485	Catherine	Abel
2	29487	Humberto	Acevedo
3	29488	Pilar	Ackeman
4	28866	Aaron	Adams
5	13323	Adam	Adams
6	29170	Alexandra	Adams
7	18504	Angel	Adams
8	13280	Bailey	Adams
9	28678	Ben	Adams
10	18075	Courtney	Adams
11	21871	Devin	Adams

Query executed successfully.

3) Utiliza las tablas *Sales.SalesOrderHeader* y *Sales.Customer* para crear una consulta que muestre el ID de la orden (SalesOrderID) y el ID del cliente (CustomerID). Asegúrate de incluir todas las órdenes, incluso si no hay un cliente asociado usando un RIGHT JOIN.

```
SELECT sso.SalesOrderID,
       sc.CustomerID
FROM Sales.SalesOrderHeader sso
     RIGHT OUTER JOIN Sales.Customer sc
     ON sc.CustomerID = sso.CustomerID
```

	SalesOrderID	CustomerID
1	NULL	1
2	NULL	2
3	NULL	3
4	NULL	4
5	NULL	5
6	NULL	6
7	NULL	7
8	NULL	8
9	NULL	9
10	NULL	10
11	NULL	11

4) Utiliza las tablas *Production.Product* y *Production.ProductReview* para crear una consulta que muestre el nombre del producto (Name) y el comentario de la reseña (Comments). Incluye todos los productos y todas las reseñas, incluso si no coinciden usando un FULL OUTER JOIN.

```
SELECT p.Name,
       pr.Comments
FROM Production.Product p
     FULL OUTER JOIN Production.ProductReview pr
     ON p.ProductID = pr.ProductID
```

	Name	Comments
1	Adjustable Race	NULL
2	Bearing Ball	NULL
3	BB Ball Bearing	NULL
4	Headset Ball Bearings	NULL
5	Blade	NULL
6	LL Crankarm	NULL
7	ML Crankarm	NULL
8	HL Crankarm	NULL
9	Chainring Bolts	NULL
10	Chainring Nut	NULL
11	Chainring	NULL

5) Utiliza las tablas *Production.Product* y *Production.ProductCategory* para crear una consulta que devuelva una combinación de todos los productos con todas las categorías usando un CROSS JOIN.

```
SELECT
    p.Name AS ProductName,
    pc.Name AS CategoryName
FROM Production.Product p
CROSS JOIN Production.ProductCategory pc
```

100 %

Results		Messages
	ProductName	CategoryName
1	Adjustable Race	Accessories
2	All-Purpose Bike Stand	Accessories
3	AWC Logo Cap	Accessories
4	BB Ball Bearing	Accessories
5	Bearing Ball	Accessories
6	Bike Wash - Dissolver	Accessories
7	Blade	Accessories
8	Cable Lock	Accessories
9	Chain	Accessories
10	Chain Stays	Accessories
11	Chainring	Accessories

6) Crea dos consultas: una que utilice UNION y otra que utilice UNION ALL para combinar los nombres de productos de las tablas *Production.Product* y *Production.ProductModel*.

```
SELECT ProductModelID,
       Name
FROM Production.Product
UNION
SELECT ProductModelID,
       Name
FROM Production.ProductModel;
```

100 %

Results		Messages
	ProductModelID	Name
1	NULL	Adjustable Race
2	NULL	BB Ball Bearing
3	NULL	Bearing Ball
4	NULL	Blade
5	NULL	Chain Stays
6	NULL	Chainring
7	NULL	Chainring Bolts
8	NULL	Chainring Nut
9	NULL	Cone-Shaped Race
10	NULL	Crown Race
11	NULL	Cup-Shaped Race

```

SELECT Name
FROM Production.Product
UNION ALL
SELECT Name

```

100 %

Results		Messages
	Name	
1	Adjustable Race	
2	All-Purpose Bike Stand	
3	AWC Logo Cap	
4	BB Ball Bearing	
5	Bearing Ball	
6	Bike Wash - Dissolver	
7	Blade	
8	Cable Lock	
9	Chain	
10	Chain Stays	
11	Chainring	

7) Utiliza la tabla *HumanResources.Employee* para crear una consulta que muestre el *BusinessEntityID*, el *JobTitle*, y una columna adicional que indique si el título del puesto contiene la palabra 'Manager' usando la expresión CASE. Si el título del puesto es nulo, usa COALESCE para mostrar 'No Title'.

```

SELECT
    e.BusinessEntityID,
    COALESCE(e.JobTitle, 'No Title') AS JobTitle,
    CASE
        WHEN e.JobTitle LIKE '%Manager%' THEN 'Manager'
        ELSE 'Not Manager'
    END AS JobCategory
FROM
    HumanResources.Employee e;

```

100 %

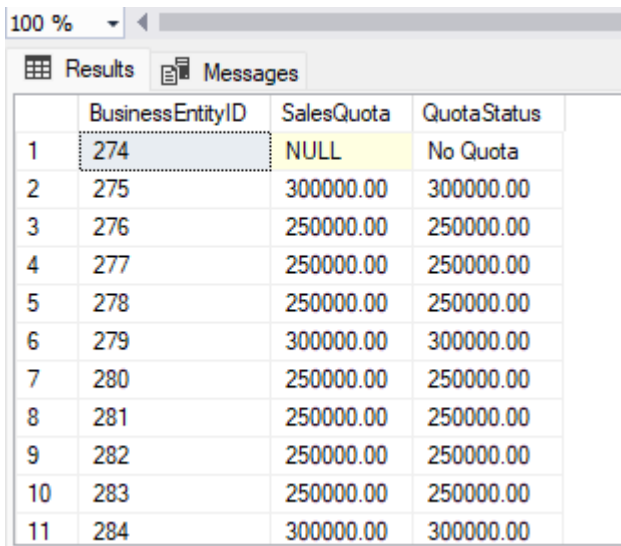
Results

Messages

	BusinessEntityID	JobTitle	JobCategory
1	1	Chief Executive Officer	Not Manager
2	2	Vice President of Engineering	Not Manager
3	3	Engineering Manager	Manager
4	4	Senior Tool Designer	Not Manager
5	5	Design Engineer	Not Manager
6	6	Design Engineer	Not Manager
7	7	Research and Development Manager	Manager
8	8	Research and Development Engineer	Not Manager
9	9	Research and Development Engineer	Not Manager
10	10	Research and Development Manager	Manager
11	11	Senior Tool Designer	Not Manager

8) Utiliza la tabla *Sales.SalesPerson* para crear una consulta que muestre el *BusinessEntityID*, la *SalesQuota*, y una columna que indique 'No Quota' si *SalesQuota* es nulo usando la función *ISNULL*.

```
SELECT sp.BusinessEntityID,  
       sp.SalesQuota,  
       ISNULL(CAST(sp.SalesQuota AS VARCHAR(50)), 'No Quota') AS QuotaStatus  
-- Se usa CAST(sp.SalesQuota AS VARCHAR(50)) para convertir SalesQuota a  
texto,  
-- ya que la función ISNULL requiere que ambos valores (el original y el  
alternativo) sean del mismo tipo de datos.  
FROM Sales.SalesPerson sp;
```



	BusinessEntityID	SalesQuota	QuotaStatus
1	274	NULL	No Quota
2	275	300000.00	300000.00
3	276	250000.00	250000.00
4	277	250000.00	250000.00
5	278	250000.00	250000.00
6	279	300000.00	300000.00
7	280	250000.00	250000.00
8	281	250000.00	250000.00
9	282	250000.00	250000.00
10	283	250000.00	250000.00
11	284	300000.00	300000.00