# **Some query examples:**

# Filtering data

**Return the three most recent orders:**

SELECT TOP (3) orderid, orderdate, custid, empid

FROM Sales.Orders

ORDER BY orderdate DESC;

**Return 1% of most recent orders.**

SELECT TOP (1) PERCENT orderid, orderdate, custid, empid

FROM Sales.Orders

ORDER BY orderdate DESC;

**Return ‘n’ number of most recent orders**

DECLARE @n AS BIGINT = 5;

SELECT TOP (@n) orderid, orderdate, custid, empid

FROM Sales.Orders

ORDER BY orderdate DESC;

**Define ordering based on order date descending, followed by order ID descending; it then skips the first 50 rows and fetches the next 25 rows:**

SELECT orderid, orderdate, custid, empid

FROM Sales.Orders

ORDER BY orderdate DESC, orderid DESC

OFFSET 50 ROWS FETCH NEXT 25 ROWS ONLY;

**Return rows from a specific page.**

DECLARE @pagesize AS BIGINT = 25, @pagenum AS BIGINT = 3;

SELECT orderid, orderdate, custid, empid

FROM Sales.Orders

ORDER BY orderdate DESC, orderid DESC

OFFSET (@pagenum - 1) \* @pagesize ROWS FETCH NEXT

@pagesize ROWS ONLY;

# Joins

**Return suppliers from Japan and the products they supply**

SELECT

S.companyname AS supplier, S.country,

P.productid, P.productname, P.unitprice

FROM Production.Suppliers AS S

INNER JOIN Production.Products AS P

ON S.supplierid = P.supplierid

WHERE S.country = N'Japan';

**Join two instances of the HR.Employees table to match employees with their managers**

SELECT E.empid,

E.firstname + N' ' + E.lastname AS emp,

M.firstname + N' ' + M.lastname AS mgr

FROM HR.Employees AS E

INNER JOIN HR.Employees AS M

ON E.mgrid = M.empid;

**To include CEO row (has NULL for mgrid)**

SELECT E.empid,

E.firstname + N' ' + E.lastname AS emp,

M.firstname + N' ' + M.lastname AS mgr

FROM HR.Employees AS E

LEFT OUTER JOIN HR.Employees AS M

ON E.mgrid = M.empid;

**Return suppliers from Japan and the products they supply, including suppliers from Japan that don’t have related products.**

SELECT

S.companyname AS supplier, S.country,

P.productid, P.productname, P.unitprice

FROM Production.Suppliers AS S

LEFT OUTER JOIN Production.Products AS P

ON S.supplierid = P.supplierid

WHERE S.country = N'Japan';

**Join query that copares NULL values as well**

SELECT EL.country, EL.region, EL.city, EL.numemps,

CL.numcusts

FROM dbo.EmpLocations AS EL

INNER JOIN dbo.CustLocations AS CL

ON EL.country = CL.country

AND ISNULL(EL.region, N'<N/A>') = ISNULL(CL.region,

N'<N/A>')

AND EL.city = CL.city;

**Return all suppliers from Japan, and matching products where relevant.**

SELECT

S.companyname AS supplier, S.country,

P.productid, P.productname, P.unitprice,

C.categoryname

FROM Production.Suppliers AS S

LEFT OUTER JOIN Production.Products AS P

ON S.supplierid = P.supplierid

LEFT OUTER JOIN Production.Categories AS C

ON C.categoryid = P.categoryid

WHERE S.country = N'Japan';

# Built in functions

**Query the input employee row, and prints a message if the requested employee was not found:**

DECLARE @empid AS INT = 10;

SELECT empid, firstname, lastname

FROM HR.Employees

WHERE empid = @empid;

IF @@ROWCOUNT = 0

PRINT CONCAT('Employee ', CAST(@empid AS VARCHAR(10)), ' was not found.');

The @@ROWCOUNT function returns an INT typed value. If the row

count can exceed the maximum INT value (2,147,483,647), use the

ROWCOUNT\_BIG function, which returns a BIGINT typed value.