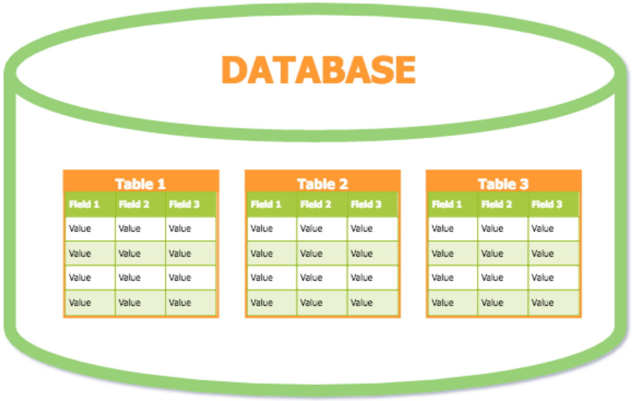
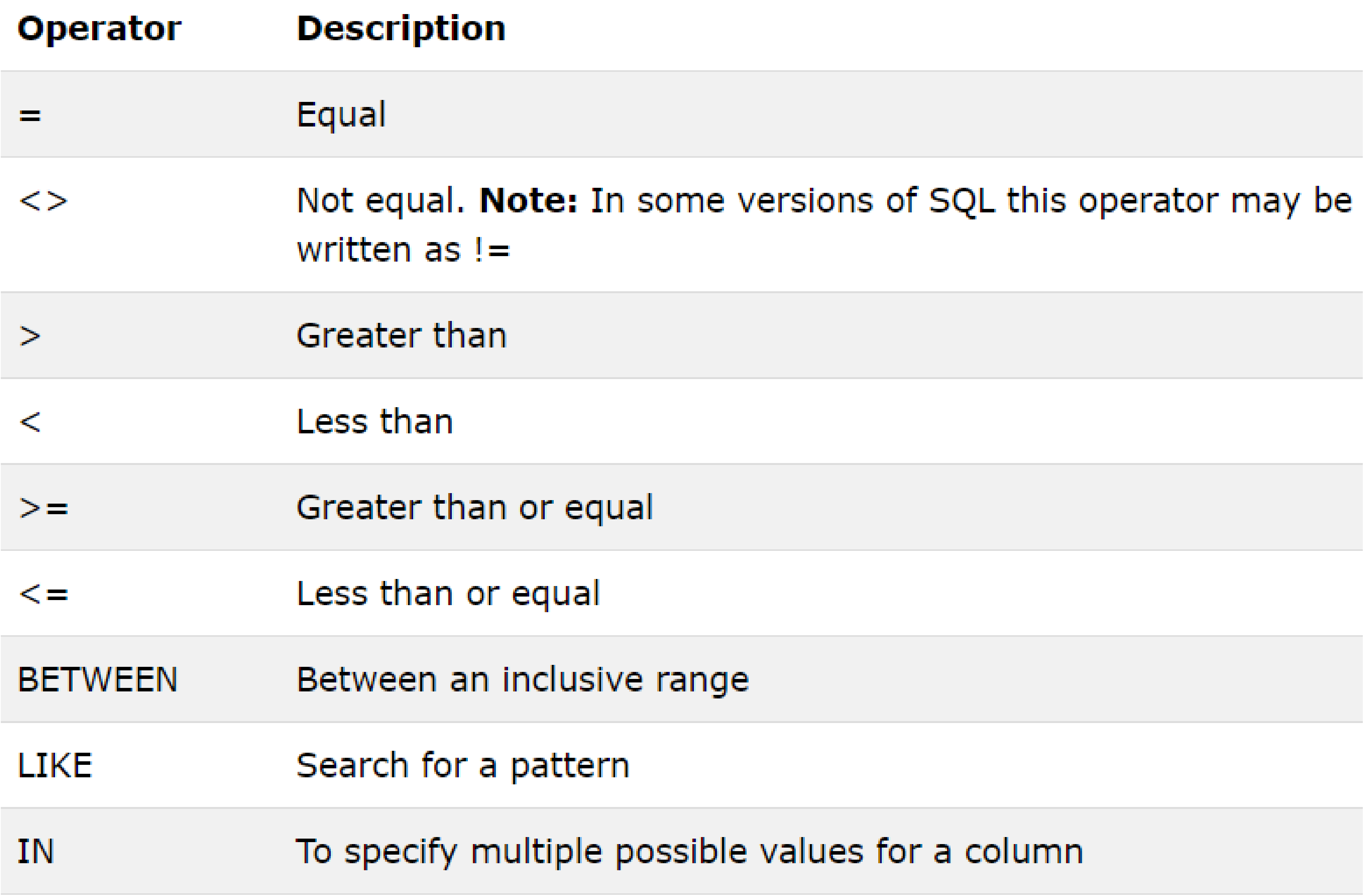
ISAD253SL - Databases

Lesson 8
Data Manipulation in SQL



Dileeka Alwis
Lecturer, School of Computing, NSBM

Operators used in WHERE Clause



Operator	**Description**
=	Equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

AND, OR and NOT Operators

- The WHERE clause is combined with these operators to filter records.
- AND and OR operators are used to filter records based on more than one condition.
- **AND** - Displays a record if **all** the conditions are **TRUE**.
- **OR** - Displays a record if **any** of the conditions are **TRUE**.
- **NOT** - Displays a record if the condition is **NOT TRUE**.

Combining AND, OR and NOT

- Can combine AND, OR and NOT operators a single WHERE clause.

- Have to use parenthesis “**( )**” to establish the order of precedence.

- If the parenthesis are not used, then the order of evaluation will be;

  1. **NOT**
  2. **AND**
  3. **OR**

BETWEEN Operator

- Used to selects values within a given range.
- The values can be numbers, text, or dates.
- The BETWEEN operator is inclusive: begin and end values are included.

BETWEEN <lowest_value> **AND** <highest_value>

LIKE Operator

- Used in a WHERE clause to search for a specified pattern in a field.

- There are two **wildcard characters** used in conjunction with the LIKE operator to substitute any other character(s) in a string.

% (Percentage sign):

Represents zero, one, or multiple characters

_ (Underscore):

Represents a single character

Character List Wildcard

- Used to define sets and ranges of characters (list of characters) to match or not match.
- [charlist] - Defines sets and ranges of characters to match
- [^charlist] or [!charlist] - Defines sets and ranges of characters **NOT** to match

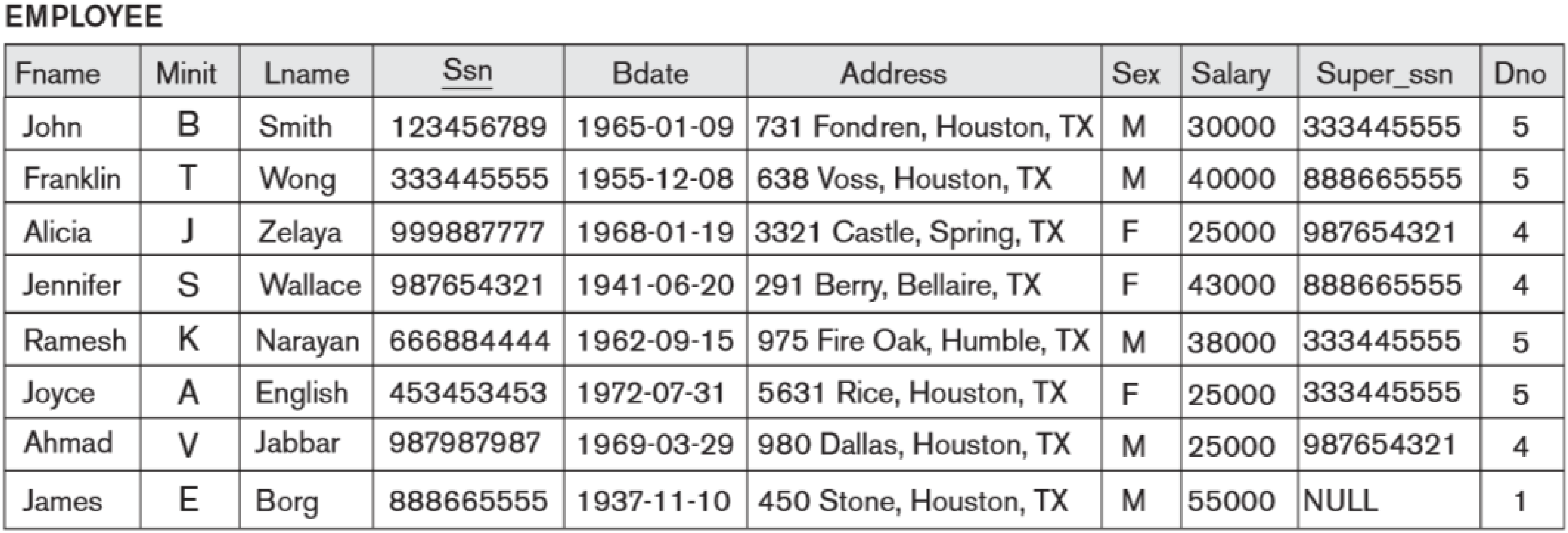
IN Operator

- Allows to specify multiple values in a WHERE clause.

- The IN operator is a shorthand for multiple OR conditions.

```
SELECT <Field names>
FROM <Table_Name>
WHERE <Field_Name>
IN (<Value1, Value2>, ...)
```

Exercise



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Exercise

- Display the ssn and first name of male employees working in department 5 and female employees working in department 4.

- Retrieve the ssn, name and DOB of employees who were born in 1960 -1970.

- Display the surnames of employees whose surname contains a vowel as the 2nd letter.

- Retrieve the names of employees whose first name is not ‘John, Alicia and Ramesh’.

ORDER BY

- Used to arrange (sort) the rows according to specific criteria.

- **ASC -** Order rows in ascending order (Default)

- **DESC -** Order rows in descending order

Order of Execution

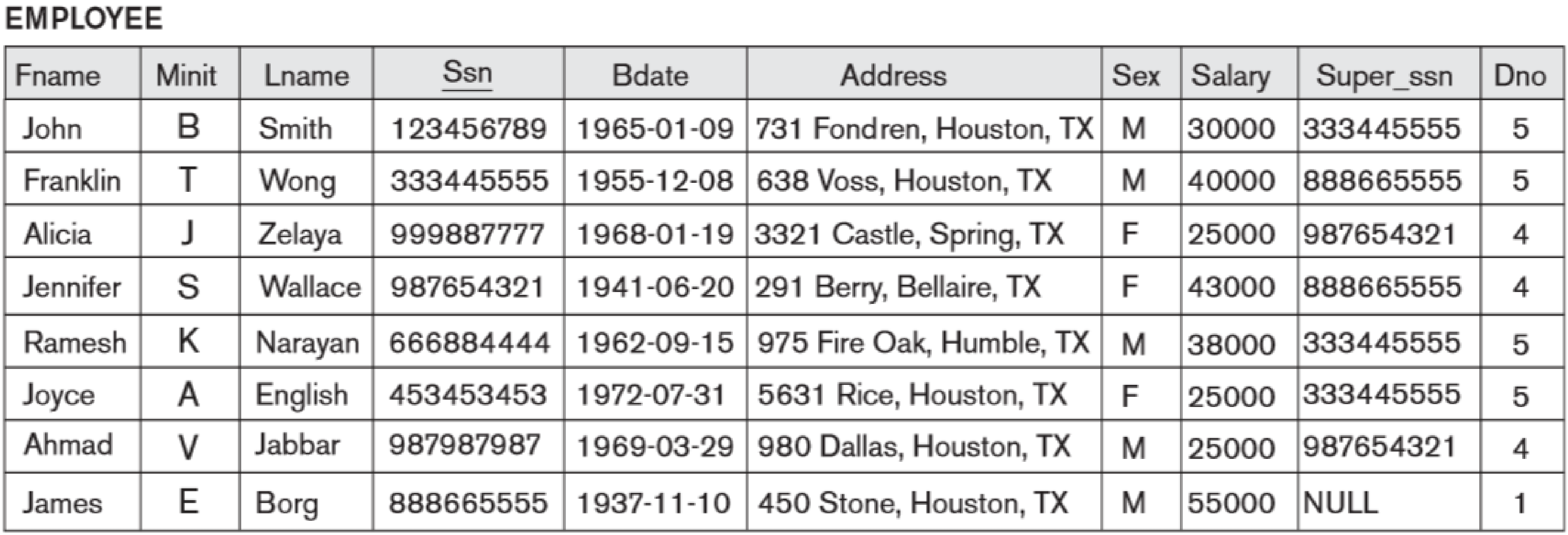
SELECT <attribute list>

FROM <table list>

WHERE <condition>

ORDER BY <attribute list>

Exercise



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

- Display all details in ascending order of surname.
- Display all details in ascending order of salary and descending order of surname.

GROUP BY

- This clause is used to group the result-set by one or more columns.

- Often used with aggregate functions.

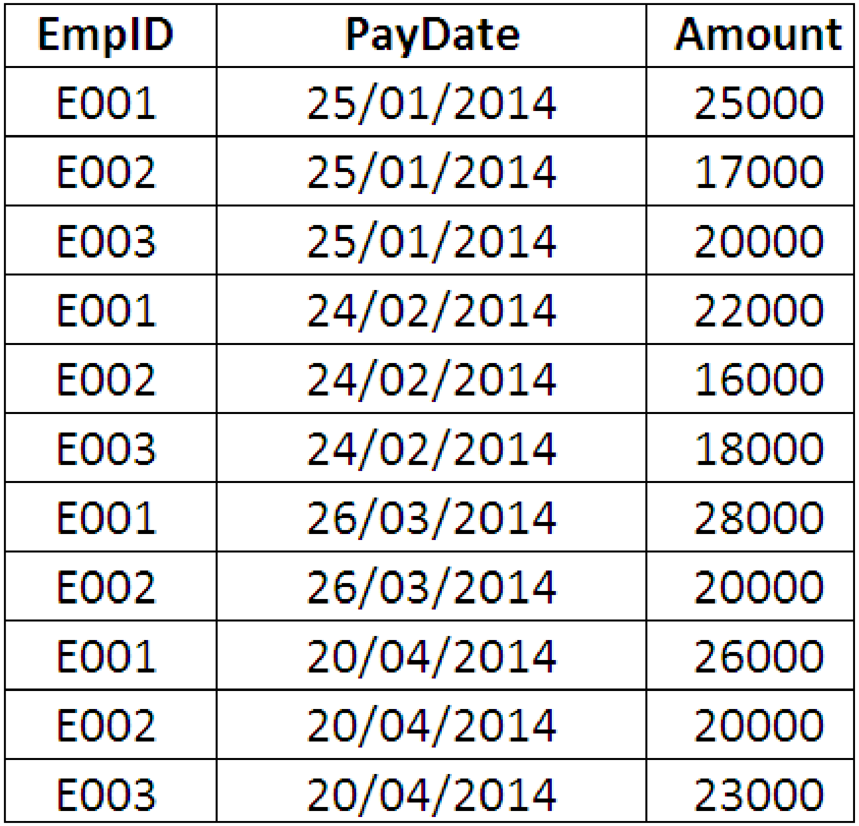
```
SELECT <Field_names>

FROM <Table_Name>

GROUP BY <Field_name>
```

Exercise

Payment Table



EmpID	PayDate	Amount
E001	25/01/2014	25000
E002	25/01/2014	17000
E003	25/01/2014	20000
E001	24/02/2014	22000
E002	24/02/2014	16000
E003	24/02/2014	18000
E001	26/03/2014	28000
E002	26/03/2014	20000
E001	20/04/2014	26000
E002	20/04/2014	20000
E003	20/04/2014	23000

- Display all the salary records of each employee together.
- Display the total amount earned by each employee.

HAVING

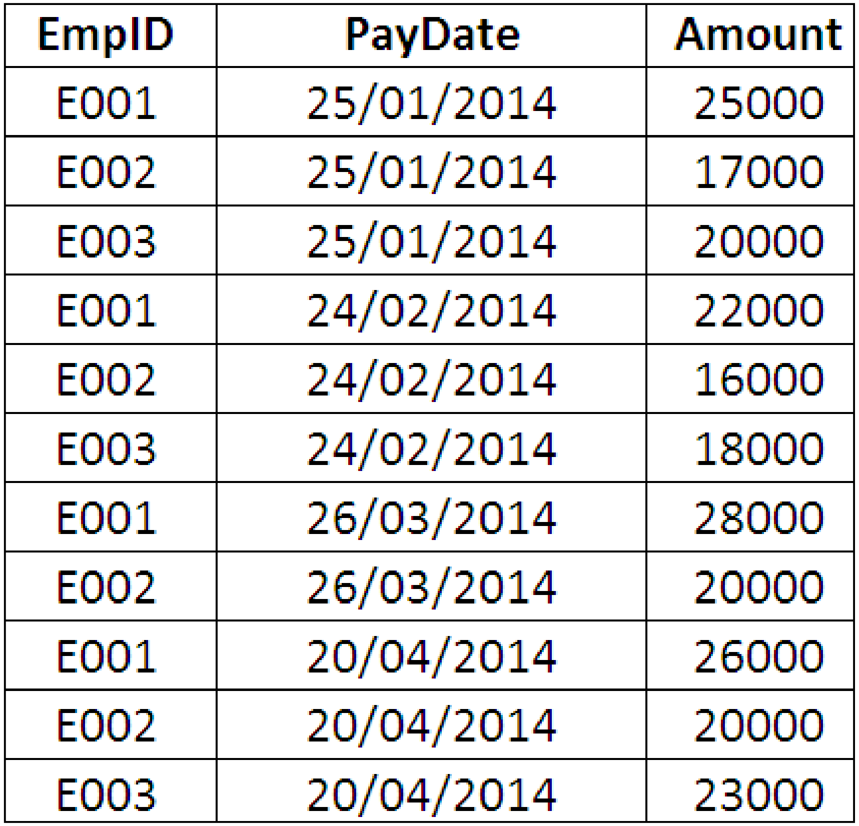
- Often used with the GROUP BY clause to apply a filter condition to the columns that appear in the GROUP BY clause.

- If the GROUP BY clause is omitted, the HAVING clause behaves like the WHERE clause.

- **Notice:**
  - HAVING clause applies the condition to each group of rows.
  - WHERE clause applies the condition to each individual row.

Exercise

Payment Table



EmpID	PayDate	Amount
E001	25/01/2014	25000
E002	25/01/2014	17000
E003	25/01/2014	20000
E001	24/02/2014	22000
E002	24/02/2014	16000
E003	24/02/2014	18000
E001	26/03/2014	28000
E002	26/03/2014	20000
E001	20/04/2014	26000
E002	20/04/2014	20000
E003	20/04/2014	23000

- Display the maximum salary record of employee ‘E002’.
- Display only the records with amount greater than 20000 in descending order of EmpID .

Order of Execution

SELECT <attribute and function list>

FROM <table list>

WHERE <condition>

GROUP BY <grouping attribute(s)>

HAVING <group condition>

ORDER BY <attribute list>

SQL Sub Queries

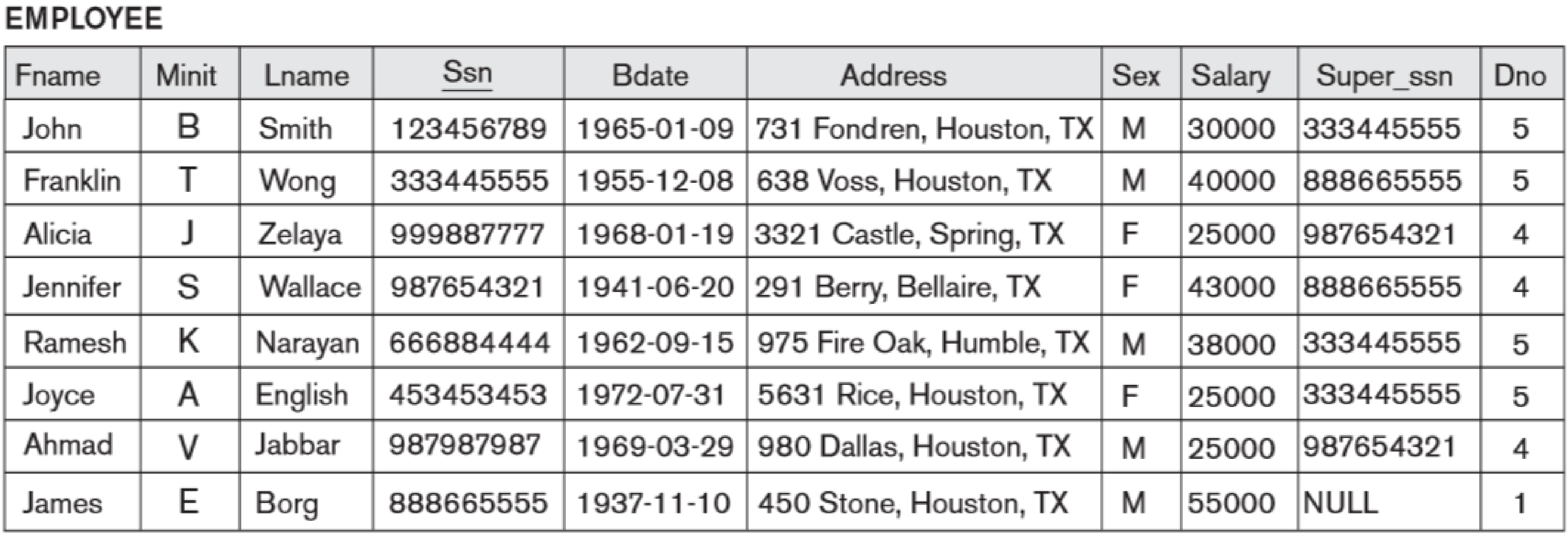
- A Sub query (Inner query/ Nested query) is a query within another SQL query and embedded within the WHERE clause.

- Enclose sub query in parentheses.

- Can’t apply on columns containing text.

- Return a single value or list of values.

Return a single value

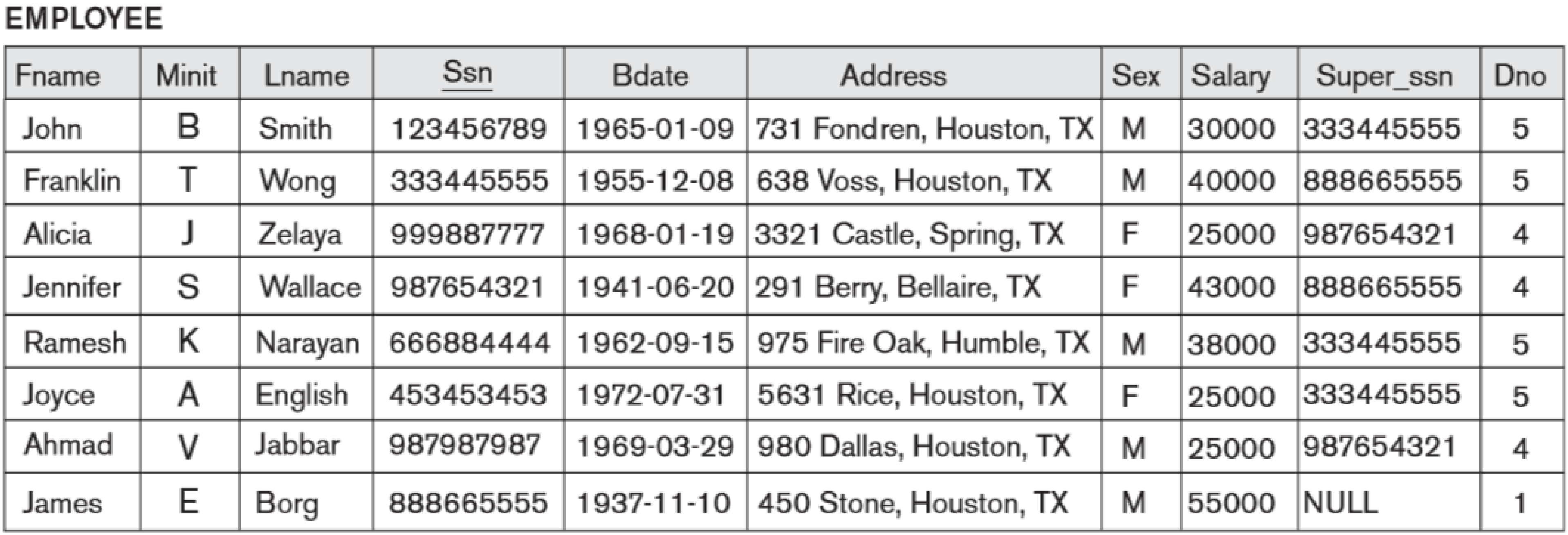


EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

- Display ssn, fname and surname of all the employee who work in the same department where employee ‘333445555’ is working.

Return a list of values



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

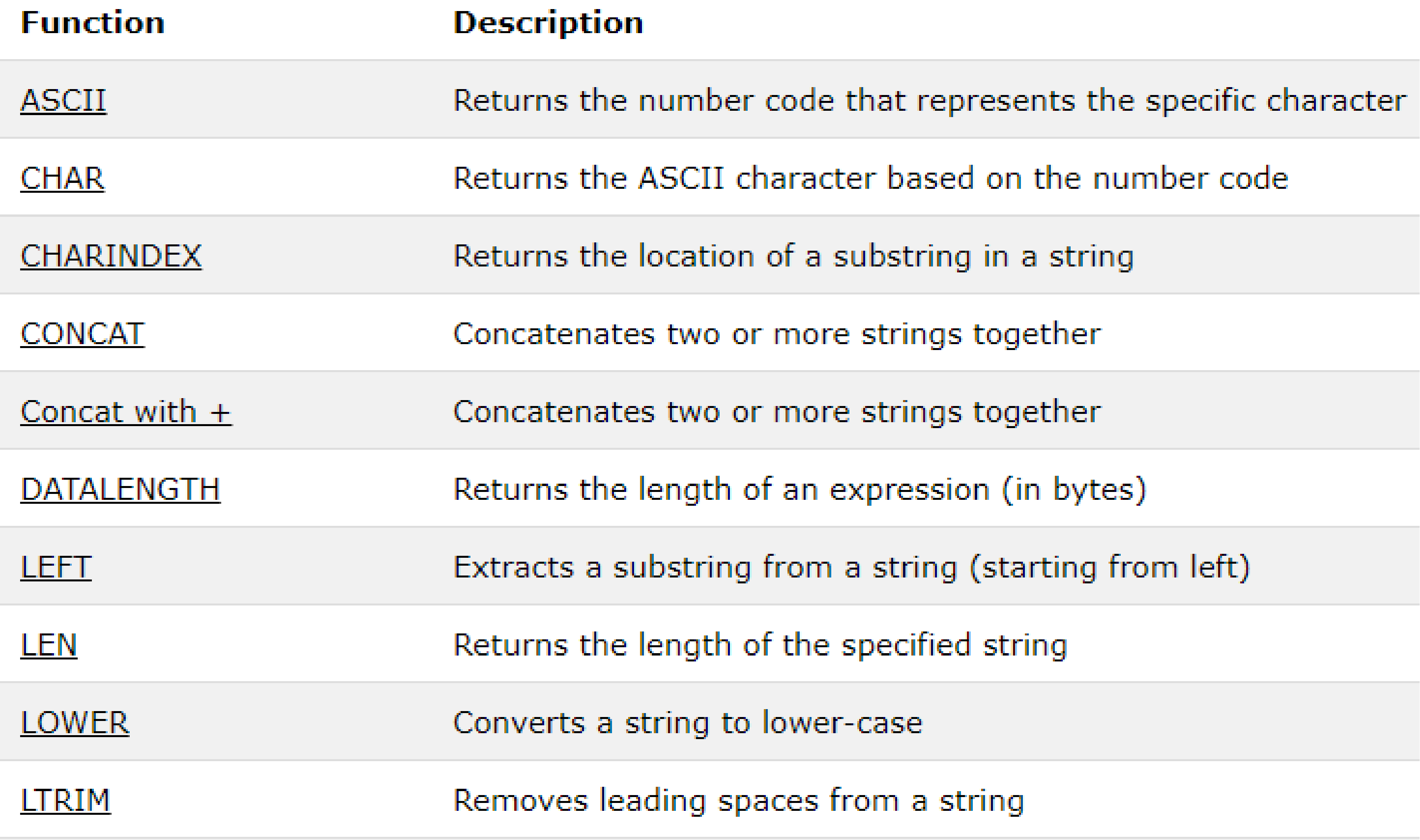
- Display ssn, fname and Department No of all the employee who work in departments located at ‘Houston’.

Built-in Functions

- SQL Server has many built-in functions.

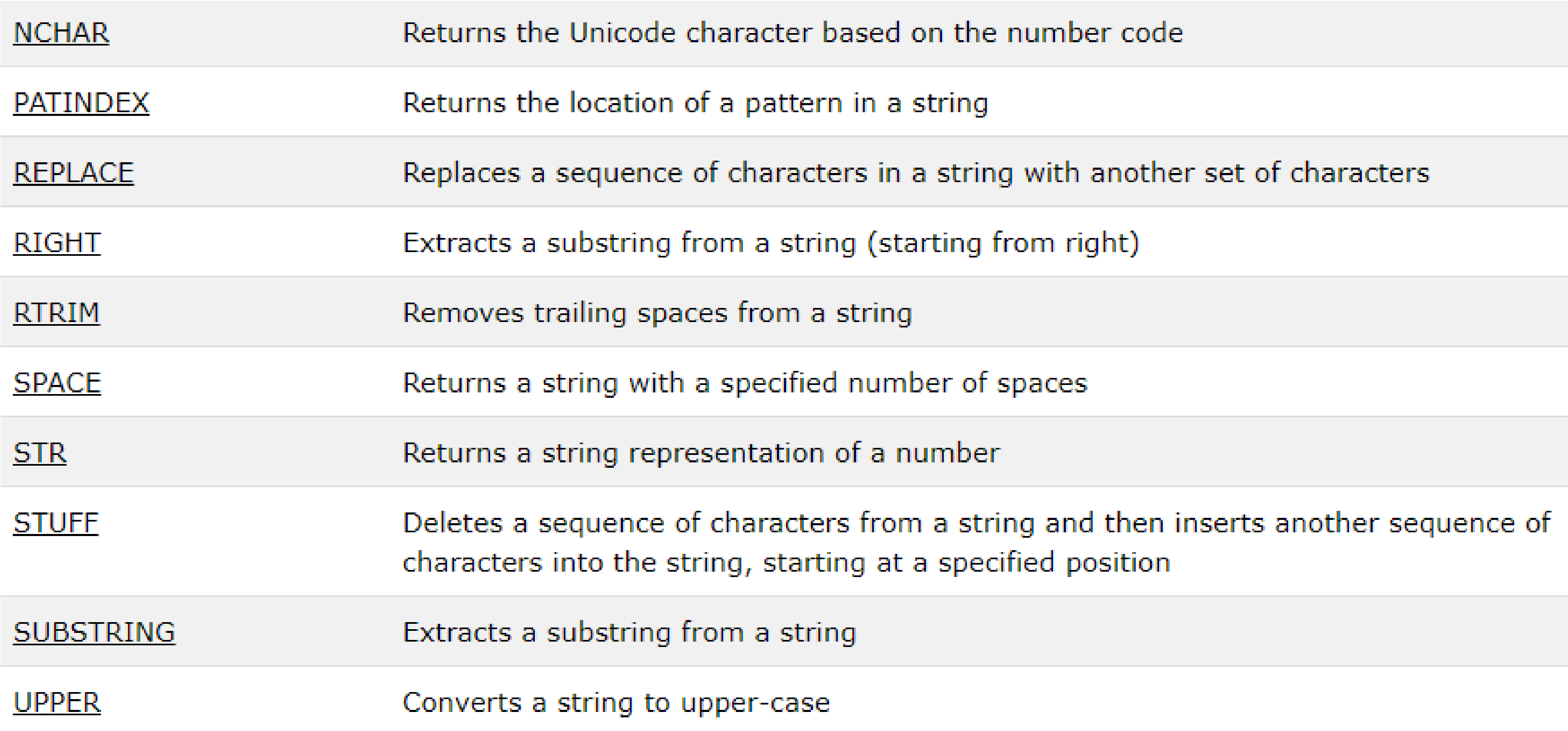
  – String functions
  – Numeric functions
  – Date functions
  – Conversion functions
  – Advanced functions

String functions



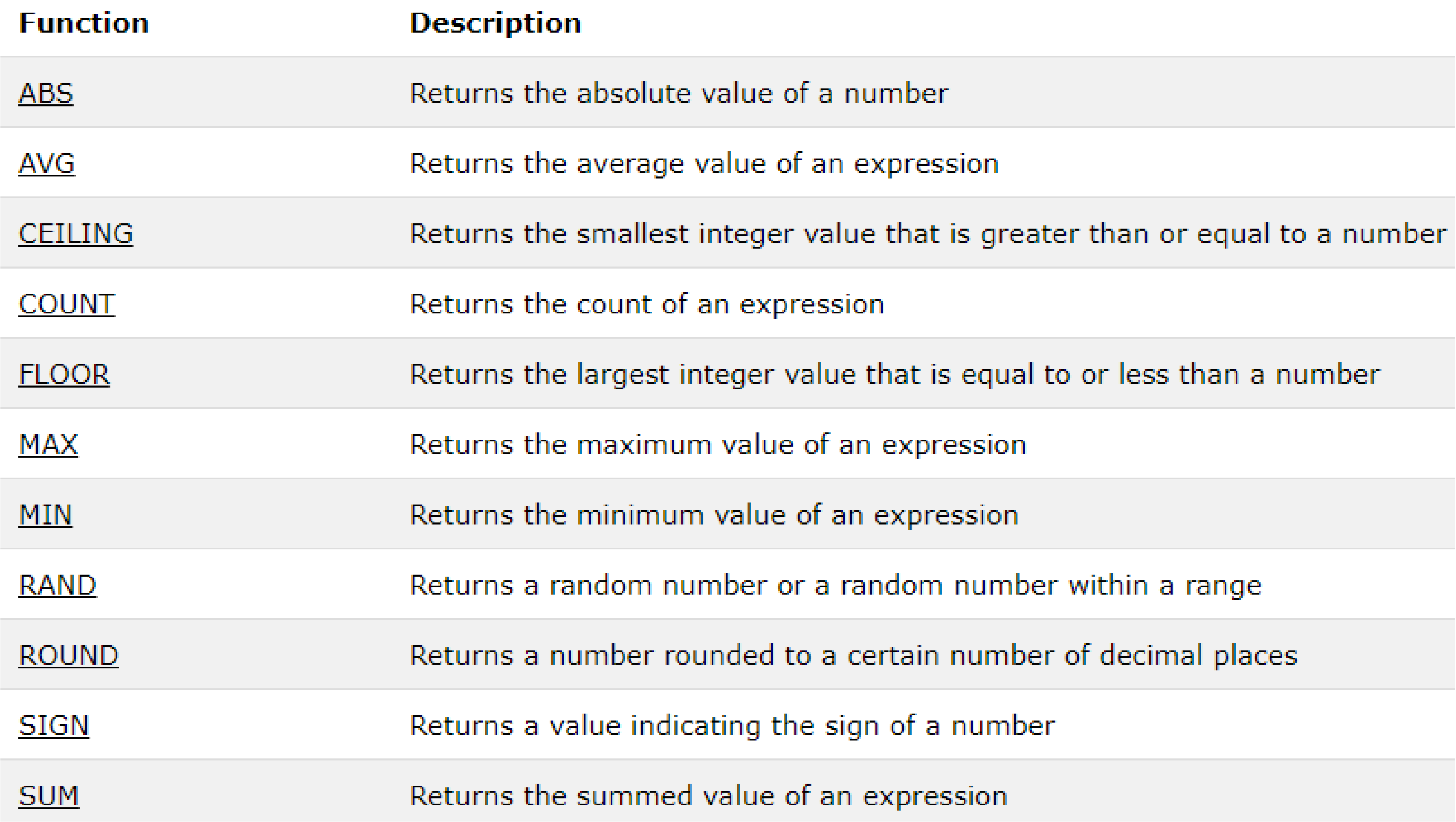
Function	Description
ASCII	Returns the number code that represents the specific character
CHAR	Returns the ASCII character based on the number code
CHARINDEX	Returns the location of a substring in a string
CONCAT	Concatenates two or more strings together
Concat with +	Concatenates two or more strings together
DATALENGTH	Returns the length of an expression (in bytes)
LEFT	Extracts a substring from a string (starting from left)
LEN	Returns the length of the specified string
LOWER	Converts a string to lower-case
LTRIM	Removes leading spaces from a string

String functions



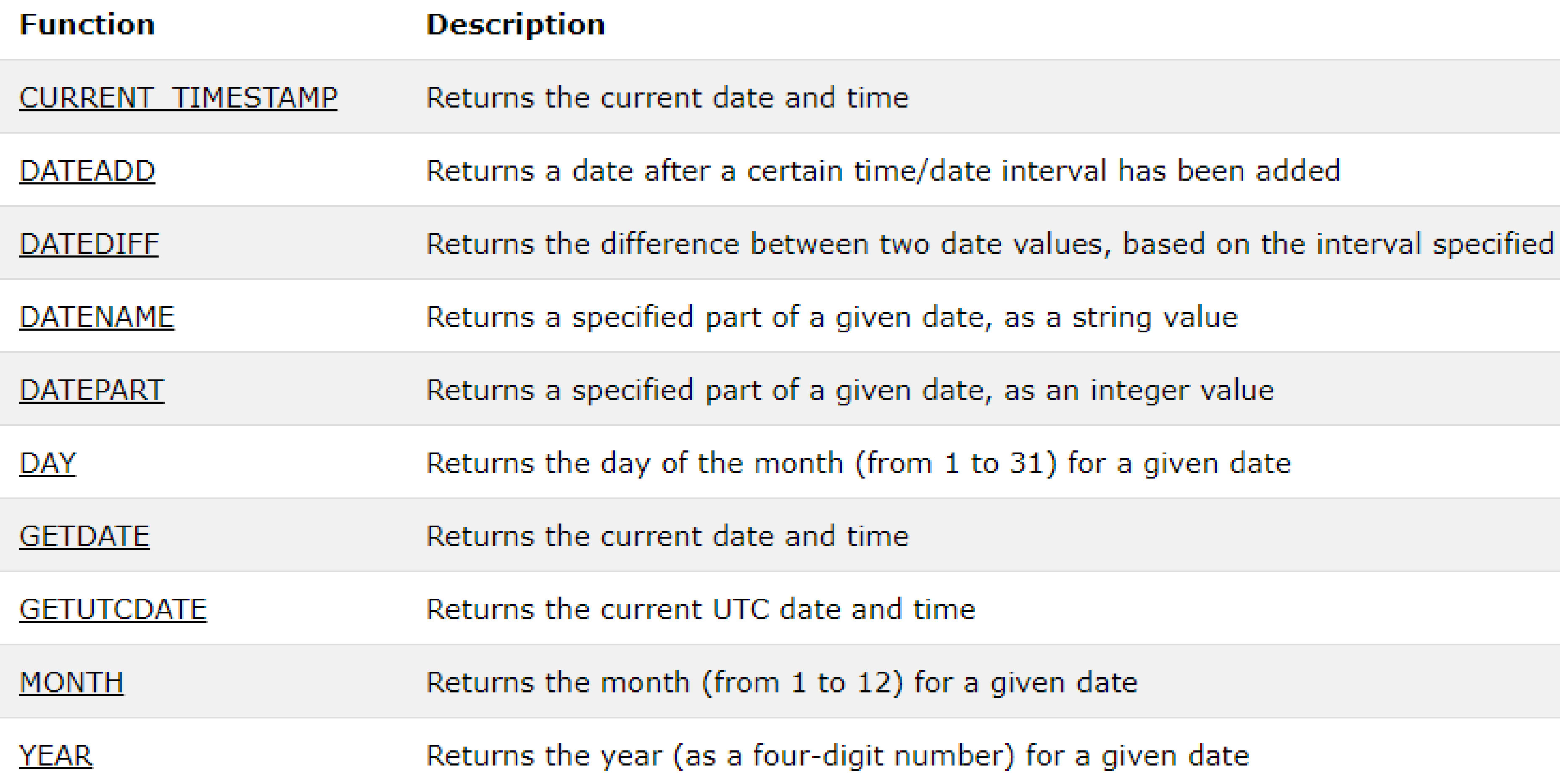
NCHAR	Returns the Unicode character based on the number code
PATINDEX	Returns the location of a pattern in a string
REPLACE	Replaces a sequence of characters in a string with another set of characters
RIGHT	Extracts a substring from a string (starting from right)
RTRIM	Removes trailing spaces from a string
SPACE	Returns a string with a specified number of spaces
STR	Returns a string representation of a number
STUFF	Deletes a sequence of characters from a string and then inserts another sequence of characters into the string, starting at a specified position
SUBSTRING	Extracts a substring from a string
UPPER	Converts a string to upper-case

Numeric Functions



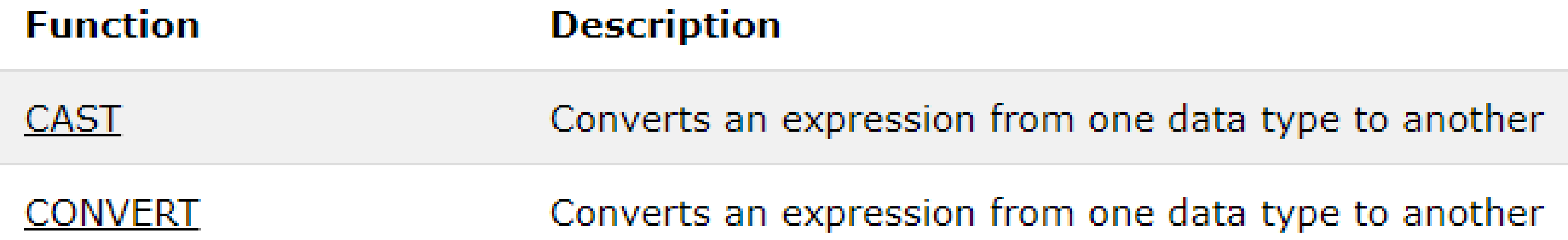
Function	Description
ABS	Returns the absolute value of a number
AVG	Returns the average value of an expression
CEILING	Returns the smallest integer value that is greater than or equal to a number
COUNT	Returns the count of an expression
FLOOR	Returns the largest integer value that is equal to or less than a number
MAX	Returns the maximum value of an expression
MIN	Returns the minimum value of an expression
RAND	Returns a random number or a random number within a range
ROUND	Returns a number rounded to a certain number of decimal places
SIGN	Returns a value indicating the sign of a number
SUM	Returns the summed value of an expression

Date Functions



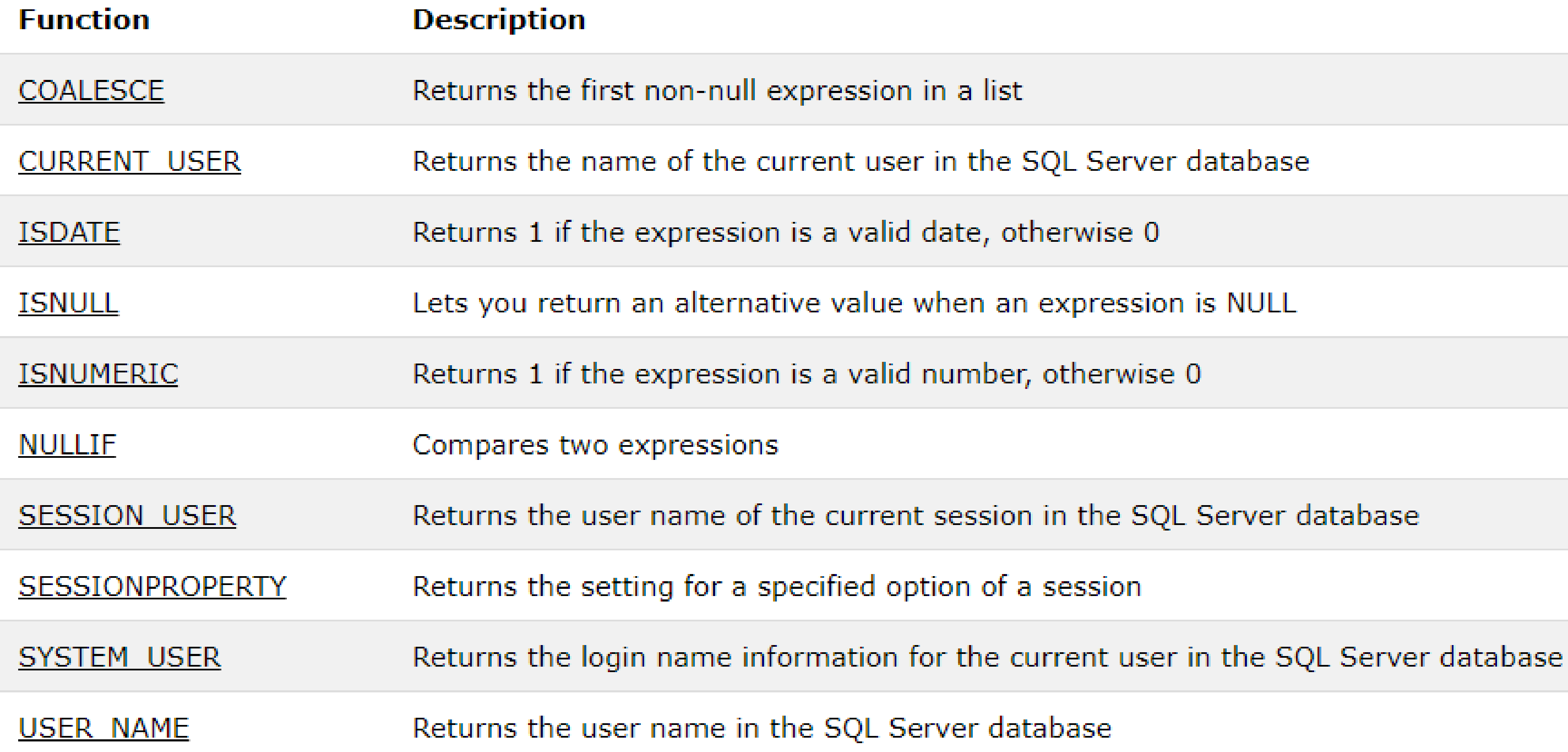
Function	Description
CURRENT_TIMESTAMP	Returns the current date and time
DATEADD	Returns a date after a certain time/date interval has been added
DATEDIFF	Returns the difference between two date values, based on the interval specified
DATENAME	Returns a specified part of a given date, as a string value
DATEPART	Returns a specified part of a given date, as an integer value
DAY	Returns the day of the month (from 1 to 31) for a given date
GETDATE	Returns the current date and time
GETUTCDATE	Returns the current UTC date and time
MONTH	Returns the month (from 1 to 12) for a given date
YEAR	Returns the year (as a four-digit number) for a given date

Conversion Functions



Function	Description
CAST	Converts an expression from one data type to another
CONVERT	Converts an expression from one data type to another

Advanced Functions



Function	Description
COALESCE	Returns the first non-null expression in a list
CURRENT_USER	Returns the name of the current user in the SQL Server database
ISDATE	Returns 1 if the expression is a valid date, otherwise 0
ISNULL	Lets you return an alternative value when an expression is NULL
ISNUMERIC	Returns 1 if the expression is a valid number, otherwise 0
NULLIF	Compares two expressions
SESSION_USER	Returns the user name of the current session in the SQL Server database
SESSIONPROPERTY	Returns the setting for a specified option of a session
SYSTEM_USER	Returns the login name information for the current user in the SQL Server database
USER_NAME	Returns the user name in the SQL Server database

Thank You