Last updated: 2022-04-21
Items with (*) have a comment (or explanation) that is shown as a tooltip when hovering the mouse over them

Please send feedback or error reports to: dbms_comparison@sql-workbench.eu

This comparison focuses on SQL features that can be used in SQL statements or self-contained SQL scripts that don't require additional software (e.g. a compiler) to be usable. Features for database administration or deployment are also not the focus of this comparison.

To get a more in-depth comparison about some of the SQL features compared here, please visit Modern SQL

☑ Oracle | ☑ Postgres | ☑ SQL Server | ☑ IBM DB2 | ☑ MySQL | ☑ MariaDB | ☑ Firebird | ☑ H2 | ☑ HSQLDB | ☑ SQLite

Feature	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Queries							
Window functions	Yes	Yes(*)DISTINCT is not supported inside a window function	Yes(*)DISTINCT is not supported inside a window function	Yes	Yes(*)Since 8.0	Yes(*)Since 10.2	Yes(*)Since 3.0
Common Table Expressions	Yes	Yes	Yes	Yes	Yes(*)Since 8.0	Yes(*)Since 10.2	Yes
CTE in a sub-query(*)Use a common table expression in a sub-query, not only as a top level query	Yes	Yes	No	No	Yes(*)Since 8.0	No	Yes
Recursive Queries	Yes	Yes	Yes	Yes	Yes(*)Since 8.0	Yes	Yes
Row constructor(*)Use of the VALUES row-constructor wherever a table reference can be used. Sometimes also called "table value constructor".	No	Yes	Yes(*)Can only be used in a FROM, not e.g. in a common table expression directly.	Yes	No	No	No
Filtered aggregates(*)Only include rows in an aggregate based on a condition: avg(salary) filter (where dept_id = 1)	No	Yes(*)Since 9.4	No	No	No	No	Yes(*)Since 4.0
PIVOT Support	Yes	No(*)The crosstab function can be used for this.	Yes	No	No	No	No
GROUP BY .. ROLLUP	Yes	Yes(*)Since 9.5	Yes	Yes	Yes	Yes	No
GROUP BY .. GROUPING SETS(*)Create multiple independent groups with a single GROUP BY query	Yes	Yes(*)Since 9.5	Yes	Yes	No	No	No
Temporal queries(*)Temporal queries allow querying the database (or a single table) to return the data as it was in the past	Yes	No	Yes(*)Since SQL Server 2016	Yes	No	Yes(*)Since 10.3	No
SELECT without a FROM clause	No	Yes	Yes	No	(Yes)(*)No WHERE clause is allowed e.g. select 42 where not exists (...);	(Yes)(*)No WHERE clause is allowed e.g. select 42 where not exists (...);	No
Parallel queries(*)The ability to distribute a single query over serveral CPUs	Yes	Yes(*)Full parallel query support since Postgres 11	Yes	Yes	No	No	No
Aggregates for strings	Yes(*)Limited to 32k	Yes	Yes(*)DISTINCT is not supported. Can not be used as a window function.	Yes	Yes	Yes	Yes

Feature	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Tuple comparison	(Yes)(*)Not supported for >, < <> or between operators	Yes	No	Yes	Yes(*)Not supported with the BETWEEN operator	Yes(*)Not supported with the BETWEEN operator	No
Tuple updates	Yes	Yes(*)Since 9.5	No	Yes	No	No	No
UPDATE with a join	No	Yes	Yes	No	Yes	Yes	No
ANSI date literals(*)Specify date or timestamps using ANSI literals, e.g. DATE '2014-01-31' or timestamp '2014-04-25 19:18:17'	Yes	Yes	No	Yes	Yes	Yes	Yes
Query variables(*)Variables that can be used inside a single query without the need to use procedural code	No	No	Yes	No	Yes	Yes	No
UNNEST(*)Convert an array into a set of rows	No	Yes	No	Yes	No	No	No
Split string to rows(*)Split a string delimited by a specific character into multiple rows (usable like a table)	No	Yes	Yes(*)Since 2016	No	No	No	No
Regular Expressions	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Comparison based on RegEx(*)Conditions with regular expressions that can be used e.g. in a WHERE clause	Yes	Yes	No	Yes(*)Since 11.1	Yes	Yes	Yes
Substring(*)Extract the part of a string value based on a RegEx	Yes	Yes	No	Yes(*)Since 11.1	Yes(*)Since 8.0	Yes(*)Since 10.0.5	Yes(*)Since 3.0
Replace(*)Replace values in a string based on a RegEx	Yes	Yes	No	Yes(*)Since 11.1	Yes(*)Since 8.0	Yes(*)Since 10.0.05	No
Constraints	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Deferred foreign key constraints(*)Define constraints that are checked only at commit time	Yes	Yes	No	No	No	No	No
Check constraints	Yes	Yes	Yes	Yes	Yes(*)Since 8.0.16	Yes(*)Since 10.2	Yes
Check constraints with sub-query	No	No	No	No	No	No	Yes
Check constraints using custom functions(*)Create a check constraint based on a user-defined function	No	Yes	Yes	Yes	No	No	Yes
Exclusion constraints(*)Constraints that prevent e.g. overlapping date ranges (WITHOUT OVERLAP in ANSI SQL)	No	Yes	No	Yes(*)Since 10.0	No	No	No
Statement based constraint evaluation	Yes	Yes	Yes	Yes	No	No	No
ON DELETE CASCADE(*)For foreign keys	Yes	Yes	(Yes)(*)Not for self-referencing FK constraints (to the same table)	Yes	Yes	Yes	Yes
ON UPDATE CASCADE(*)For foreign keys	No	Yes	(Yes)(*)Not for self-referencing FK constraints (to the same table)	No	Yes	Yes	Yes
Foreign keys using MATCH FULL(*)Define multi-column foreign keys that handle NULL values	No	Yes	No	No	No(*)The definition is accepted, but ignored	No(*)The definition is accepted, but ignored	No
Indexing	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Partial index(*)Define an index on a subset of a table	Yes(*)Through a function based index	Yes	(Yes)(*)WHERE conditions involving functions are not supported e.g. where upper(name) <> 'ARTHUR'	No	No	No	No
Descending Index(*)Define an index that is sorted descending	Yes	Yes	Yes	Yes	Yes(*)Since 8.0	No	(Yes)(*)It's not possible to mix ASC and DESC for multi-column indexes
Index on expression(*)Create an index based on an expression/function	Yes	Yes	(No)(*)Can be simulated using an index on a computed column	(Yes)(*)Not for DB2 LUW	Yes(*)Since 8.0	(No)(*)Can be simulating by indexing a computed column	(Yes)(*)Limited to a single expression. Can not be combined with additional columns

Feature	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Index using a custom function(*)Create an expression index using a custom function (written in a "SQL" procedural language)	Yes	Yes	No	Yes	No	No	No
Index include columns(*)Define an index on some columns and include other (non-indexed) columns	No	Yes(*)Since 11	Yes	Yes	No	No	No
Multi-column statistics(*)Create extended statistics storing dependencies between values in the columns of a single table	Yes	Yes	Yes	Yes	No	No	No
Clustered index(*)An index that contains the table data (index and table storage are the same)	Yes(*)Called Index Organized Table	No	Yes	Yes	Yes	Yes	No
Duplicate NULL values in unique index(*)The SQL standard requires that a unique index allows multiple `NULL` values.	No(*)Works for single-column indexes only, not for multi-column indexes	Yes	No	No	Yes(*)The behaviour depends on the storage engine being used.	Yes(*)The behaviour depends on the storage engine being used.	No
DML	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Writeable CTEs(*)Use DML statements inside a CTE	No	Yes(*)The result of a CTE can not be updated	Yes(*)The result of a CTE can be updated, but a CTE cannot use a DML statement	No	No	No	No
Multi-row INSERTs(*)Insert more than one row with a single INSERT statement	No	Yes	Yes	Yes	Yes	Yes	No
TRUNCATE table with FK(*)Truncate tables that are referenced by other tables	Yes(*)Oracle 12.1 introduced the `cascade` option for `truncate` which requires the FK to be defined as `on delete cascade`	Yes	No	No	No	No	No
Read consistency during DML operations(*)During a DML operation reading a column value should return the value that was valid before the statement started	Yes	Yes	Yes	Yes	No	Yes(*)Since 10.3 Requires non-default sql-mode	Yes
Use target table in sub-queries(*)Use the targe table of an UPDATE, DELETE or INSERT statement in a sub-select	Yes	Yes	Yes	Yes	No	No	Yes
MERGE(*)Update rows if they exist, insert if not	Yes	Yes(*)Using `insert ... on conflict`	Yes	Yes	Yes(*)Using `INSERT .. ON DUPLICATE`	Yes(*)Using `INSERT .. ON DUPLICATE`	Yes
SELECT .. FOR UPDATE NOWAIT(*)Select one (or more) rows and lock them for a future update. Fail with an error if the lock cannot be obtained	Yes	Yes	No(*)`FOR UPDATE` can only be used with cursors, not plain SELECT statements	No	Yes(*)Since 8.0	No	No
RETURNING clause as a result set	No	Yes	Yes	No	No	No	Yes
Parallel DML(*)Use multiple threads/workers for a single DML statement	Yes	No	No	No	No	No	No
Data Types(*)Data types that can be used for the column of a table	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
User defined datatypes(*)Crate UDTs using SQL and use those UDTs as a column's data type	Yes	Yes	No(*)SQL Server does have user defined datatypes but they can not be used for columns in a table	Yes	No	No	No
Domains(*)A special kind of user defined data type that can also include check constraints, usually based on a base data type	No	Yes	(Yes)(*)This can be done using rules (together with user defined types), but they are deprecated	No	No	No	Yes
Distinct types(*)User defined types which cannot be compared. E.g. prevent comparing a `product_id` to a `customer_id`	No	No	No	Yes	No	No	No

Feature	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Arrays	No	Yes	No	No	No	No	(Yes)(*)There is no support for arrays in SQL or JDBC. They can only be used in stored procedures.
Enums(*)De-normalize lookup values by specifying a fixed set of allowed values (a special case of a check constraint)	No	Yes	No	No	Yes	Yes	No
IP address	No	Yes	No	No	No	No	No
BOOLEAN(*)Standard boolean data type as defined by the SQL standard (usable as a column data type)	No(*)Only PL/SQL supports boolean	Yes	No(*)The `BIT` is a number type that is limited to `0` and `1`.	Yes(*)Since 11.1	No(*)MySQL's `BOOLEAN` is only a synonym for `TINYINT`	No(*)MariaDB's `BOOLEAN` is only a synonym for `TINYINT`	Yes(*)Since 3.0
Interval	Yes	Yes	No	No	No	No	No
TIME(*)A data type that only stores a time	No	Yes	Yes	Yes	Yes	Yes	Yes
DATE(*)A data type that only stores a date (without a time)	No(*)Oracle's `DATE` type stores date *and* time	Yes	Yes	Yes	Yes	Yes	Yes
TIMESTAMP(*)A data type that stores a date and time	Yes	Yes	Yes(*)The data type is named `datetime` or `datetime2`. `timestamp` is something different.	Yes	Yes(*)`TIMESTAMP` has a very limited range: from 1970 up to 2038.	Yes(*)`TIMESTAMP` has a very limited range: from 1970 up to 2038.	Yes
TIME ZONE Support(*)Support for time zones (with TIMESTAMP values)	Yes	Yes	Yes(*)The data type is called `datetimeoffset`	No	No	No	Yes(*)Since 4.0
Range types(*)A data type that represents a range of values, e.g.: all values from 1 through 100 The dates from 2014-01-01 to 2014-01-08	(No)(*)The `PERIOD FOR` introduced in 12c is something similar.	Yes	No	No	No	No	No
UUID(*)A dedicated data type for UUID storage	No	Yes	Yes	No	No	Yes(*)Since 10.7.0	No
DDL	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Transactional DDL(*)The ability to rollback any DDL statement	No	Yes	Yes	Yes	No	No	Yes
Computed columns(*)Define a column in a table that is always calculated based on other columns	Yes	Yes(*)Since 12	Yes	Yes	Yes(*)Since 5.7	Yes(*)Since 5.2	Yes
Functions as column default(*)Use *any* function (including user defined functions) as the default for a column	(Yes)(*)Only built-in functions can be used. No PL/SQL functions	Yes	Yes	No	No	Yes(*)Since 10.2	Yes(*)Since 3.0
Sequences	Yes	Yes	Yes	Yes	No	Yes(*)Since 10.3	Yes
Auto increment columns(*)Columns that are populated automatically with unique values without the usage of triggers	Yes(*)Since 12c	Yes	Yes	Yes	Yes	Yes	Yes
IDENTITY columns(*)Standard compliant IDENTITY columns	(Yes)(*)Does not support OVERRIDING SYSTEM VALUE	Yes(*)Since 10	No(*)SQL Server's `identity` columns are not compatible with the SQL standard.	(Yes)(*)Does not support OVERRIDING SYSTEM VALUE	No	No	Yes(*)Since 3.0
Synonyms	Yes	No	Yes	Yes	No	No	No
Non-blocking index creation(*)Create an index without blocking DML on the table	Yes	Yes	Yes	Yes	No	No	No
Partitioning	Yes	(Yes)(*)It's not possible to create globally unique indexes on partitioned tables without including the partition key	Yes	Yes	(Yes)(*)It's not possible to create globally unique indexes on partitioned tables without including the partition key	(Yes)(*)It's not possible to create globally unique indexes on partitioned tables without including the partition key	No

Feature	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Cascading DROP(*)Drop a table including incoming foreign keys	Yes	Yes	No	Yes	No(*)MySQL accepts the `CASCADE` keyword but silently ignores it.	No(*)MySQL accepts the `CASCADE` keyword but silently ignores it.	No
DDL Triggers(*)Define triggers that are fired when a DDL statement is executed	Yes	Yes	Yes	No	No	No	Yes(*)Since 3.0
TRUNCATE Trigger(*)Define triggers that are fired when a TRUNCATE statement is executed	(No)(*)Possible through a system trigger	Yes	No	No	No	No	No
Custom name for PK constraint(*)Specify the name of the PK constraint	Yes	Yes	Yes	Yes	No	No	Yes
ALTER a table used in a view(*)The ability to `ALTER` the definition of a table used in a view without dropping the view	Yes	No(*)You can add columns to a table but not drop columns or change their data type	Yes	Yes	Yes	Yes	Yes
Add table column at specific position(*)Add a new column to a table at any position rather then only appending it to the end. This is used to influence the column order for `select *` queries	No	No	No	No	Yes	Yes	Yes
Materialized views(*)Persist the result of a query as a table	Yes	Yes	Yes(*)Called indexed views	Yes(*)Called materialized query tables (MQT)	No	No	No
MVIEW with query rewrite(*)If applicable can the optimizer detect that a query against the base tables can be done using the materialized view	Yes	No	Yes	No	No	No	No
Automatically updated MVIEWS(*)MVIEW is automatically updated when the underlying tables are changed	Yes	No	Yes	Yes	No	No	No
Temporary Tables	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Permanent global temporary tables(*)Tmporary tables that are created once and that need to be dropped manualy (their definition is retained across server restarts)	Yes	No	No	Yes	No	No	Yes
Global temporary tables(*)Temporary tables that are always visible, but the data is session specific	No	No	Yes	No	No	No	No
Session local temporary tables(*)Temporary tables that are automatically dropped at the end of the transaction or when the session is disconnected	No	Yes	Yes	No	Yes	Yes	No
Use a temporary table twice in a single query	Yes	Yes	Yes	Yes	No	Yes(*)Since 10.2.1	Yes
Programming	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Stored procedures(*)Writing and managing stored procedures using SQL commands	Yes	Yes(*)Procedures only since Postgres 11	Yes	Yes	Yes	Yes	Yes
Table functions(*)Functions that return result sets and can be used like a table	Yes	Yes	Yes	Yes	No	No	Yes
Custom aggregates(*)Create aggregate functions using SQL	Yes	Yes	No(*)Can be done with CLR functions	No(*)Can be done with host languages (e.g. Java)	No	No	No
Function overloading(*)Create different versions of the same function that are distinguished by their argument list	Yes(*)Only inside packages	Yes	No	Yes	No	No	No
User defined operators(*)Create new (comparison) operators for user defined data types	No(*)`CREATE OPERATOR` only creates functions. Creating operators like `=`, `<`, `>` is not possible	Yes	No	No	No	No	No

Feature	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Statement level triggers(*)Triggers that fire once for each statement	Yes	Yes	Yes	Yes	No	No	No
Row level triggers(*)Triggers that fire once for each row	Yes	Yes	No	Yes	Yes	Yes	Yes
RETURNING clause in a programming language(*)Use a `RETURNING` clause from within a programming language	Yes	Yes	Yes	No	No	No	Yes
Before triggers(*)Triggers that are fired before the changes of a DML statement are persisted	Yes	Yes	(No)(*)An INSTEAD OF trigger can be used for a similar purpose	Yes	Yes	Yes	Yes
Dynamic SQL in functions(*)The ability to use dynamic SQL in stored functions	Yes	Yes	No(*)Possible in CLR functions	Yes	No	No	Yes
Dynamic SQL in triggers(*)The ability to use dynamic SQL in triggers	Yes	Yes	Yes	No	No	No	Yes
Delete triggers fired by cascading deletes(*)When rows are deleted due to a `ON DELETE CASCADE` foreign key, are delete triggers fired?	Yes	Yes	Yes	Yes	No	No	Yes
Built-in scheduler	Yes	No	Yes	Yes	Yes	Yes	No
Views	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Updateable Views	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WITH CHECK OPTION(*)Create updateable views where only rows can be updated/deleted/inserted that match the `WHERE` clause of the view	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Triggers on views	Yes	Yes	Yes	Yes	No	No	Yes
Views with derived tables(*)Create a view that uses a derived table	Yes	Yes	Yes	Yes	No	No	Yes
JOINs and Operators	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
CROSS JOIN	Yes	Yes	Yes	Yes	Yes	Yes	Yes
FULL OUTER JOIN	Yes	Yes	Yes	Yes	No	No	Yes
LATERAL JOIN	Yes(*)Since 12c	Yes	(Yes)(*)Called `APPLY` Inner joins are not supported. Lateral joins against a derived table are not supported	Yes	Yes(*)Since 8.0.14	No	Yes(*)Since 4.0
JOIN ... USING (...)(*)A shortcut notation for the `JOIN` operator when both columns have the same name.	Yes	Yes	No	No	Yes	Yes	Yes
JOINs using tuple comparison(*)Use tuples in JOIN conditions	Yes	Yes	No	Yes	Yes	Yes	No
INTERSECT	(Yes)(*)Does not support `INTERSECT ALL`	Yes	(Yes)(*)Does not support `INTERSECT ALL`	Yes	No	Yes(*)Since 10.3	No
EXCEPT	(Yes)(*)Called `MINUS` in Oracle, but does not support the `ALL` option	Yes	(Yes)(*)Does not support `EXCEPT ALL`	Yes	No	Yes(*)Since 10.3	No
ORDER BY ... NULLS LAST	Yes	Yes	No	Yes	No	No	Yes
IS DISTINCT FROM	No	Yes	No	(Yes)(*)Not supported by DB2 LUW	Yes(*)Using the operator `<=>`	Yes(*)Using the operator `<=>`	Yes
BETWEEN SYMMETRIC	No	Yes	No	No	No	No	No
OVERLAPS(*)Checks for overlapping intervals, e.g.: `(date '2014-01-01', date '2014-09-01') overlaps (date '2014-04-01', date '2014-05-01')`	(Yes)(*)Oracle supports the `OVERLAPS` operator, but this is undocumented	Yes	No	Yes(*)Since 11.1	No	No	No
Partitioned outer join(*)A JOIN operator that can be used to fill gaps in sparse data, mainly time series. (This is unrelated to "partion wise joins" between two partitioned tables)	Yes	No(*)Since For a single time series this can be simulated using generate_series()	No	No	No	No	No
Other	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird

Feature	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
Catalogs ("databases")	(Yes)(*)Pluggable databases are available since 12.1, cross-database queries are not supported	(Yes)(*)Queries between different database (=catalogs) are not possible	Yes	No	Yes	Yes	(Yes)(*)Queries between different database (=catalogs) are not possible
Schemas	Yes	Yes	Yes	Yes	No	No	No
INFORMATION_SCHEMA(*)Support for the INFORMATION_SCHEMA defined in the SQL standard	No	Yes	Yes	No	Yes	Yes	No
NoSQL Features	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
XML Support(*)Support for a validating XML data type	Yes	Yes	Yes	Yes	Yes	Yes	No
XPath(*)Support for a XPath expressions on XML data	Yes	Yes	Yes	Yes	Yes	Yes	No
XQuery	Yes	No	Yes	Yes	No	No	No
JSON(*)Support for a (validating) JSON data type and corresponding functions	Yes(*)Introduced in version 12.1.0.**2**	Yes	Yes(*)Since SQL Server 2016	Yes(*)Since 11.5	Yes(*)Since 5.7	Yes(*)No JSON data type, only JSON functions	No
SQL/JSON Path(*)Support for SQL/JSON and JSON Path queries	Yes(*)Since 18	Yes(*)Uses non-standard function names, but supports the full JSON path syntax	Yes(*)Since 2016	No	No	No	No
Indexes on JSON documents(*)Create an index on a JSON column (and the complete JSON value) to support arbitrarty queries for elements inside the JSON value	Yes(*)Since 12.2	Yes	(No)(*)It's possible to create computed columns that exctracts a single value and index that computed column.	Yes	(No)(*)It's possible to create computed columns that exctracts a single value and index that computed column.	(No)(*)It's possible to create computed columns that exctracts a single value and index that computed column.	No
Key/Value storage	No	Yes	No	No	No(*)A key/value store is available through the "Memcached API" but it's not usable in SQL	No	No
Security	Oracle	Postgres	SQL Server	IBM DB2	MySQL	MariaDB	Firebird
User groups / Roles	Yes	Yes	Yes	Yes	Yes(*)Since 8.0	Yes(*)Since 10.0.5	Yes
Row level security(*)Allow access to the data on row level based on rules for each (database) user	Yes	Yes(*)Since 9.5	Yes	Yes	No	No	No
Grant on column level(*)Grant access to only some of the columns of a table	(Yes)(*)Not possible for SELECT grant	Yes	Yes	Yes	Yes	Yes	Yes