

## Blesková referenční příručka SQL

(POZOR jedná se o velmi zjednodušenou verzi, plné znění najdete na [www.w3schools.com](http://www.w3schools.com))

### DSL (Data Selection) část:

SQL Příkaz	Syntaxe
AND / OR	SELECT sloupec_jméno(a) FROM tabulka_jméno WHERE podmínka AND OR podmínka
AS (alias)	SELECT sloupec_jméno AS sloupec_alias FROM tabulka_jméno  <i>nebo</i>  SELECT sloupec_jméno FROM tabulka_jméno AS tabulka_alias
GROUP BY	SELECT sloupec_jméno, agregační_funkce(sloupec_jméno) FROM tabulka_jméno WHERE sloupec_jméno operátor hodnota GROUP BY sloupec_jméno
HAVING	SELECT sloupec_jméno, agregační_funkce(sloupec_jméno) FROM tabulka_jméno WHERE sloupec_jméno operátor hodnota GROUP BY sloupec_jméno HAVING agregační_funkce(sloupec_jméno) operátor hodnota
INNER JOIN	SELECT sloupec_jméno(a) FROM tabulka_jméno1 [INNER] JOIN tabulka_jméno2 ON tabulka_jméno1.sloupec_jméno=tabulka_jméno2.sloupec_jméno  <b>Oracle:</b> SELECT sloupec_jméno(a) FROM tabulka_jméno1, tabulka_jméno2 WHERE tabulka_jméno1.sloupec_jméno1 = tabulka_jméno2. sloupec_jméno2
LEFT JOIN	SELECT sloupec_jméno(a) FROM tabulka_jméno1 LEFT JOIN tabulka_jméno2 ON tabulka_jméno1.sloupec_jméno=tabulka_jméno2.sloupec_jméno  <b>Oracle:</b> SELECT sloupec_jméno(a) FROM tabulka_jméno1, tabulka_jméno2 WHERE tabulka_jméno1.sloupec_jméno1 = tabulka_jméno2. sloupec_jméno2 (+)
RIGHT JOIN	SELECT sloupec_jméno(a) FROM tabulka_jméno1 RIGHT JOIN tabulka_jméno2 ON tabulka_jméno1.sloupec_jméno=tabulka_jméno2.sloupec_jméno  <b>Oracle:</b> SELECT sloupec_jméno(a) FROM tabulka_jméno1, tabulka_jméno2 WHERE tabulka_jméno1.sloupec_jméno1 (+) = tabulka_jméno2. sloupec_jméno2
FULL JOIN	SELECT sloupec_jméno(a) FROM tabulka_jméno1

	FULL JOIN tabulka_jméno2 ON tabulka_jméno1.sloupec_jméno=tabulka_jméno2.sloupec_jméno  <b>Oracle:</b> <a href="#">nepodporováno</a>
LIKE	SELECT sloupec_jméno(a) FROM tabulka_jméno WHERE sloupec_jméno LIKE šablona/příklad
ORDER BY	SELECT sloupec_jméno(a) FROM tabulka_jméno ORDER BY sloupec_jméno [ASC DESC]
SELECT	SELECT sloupec_jméno(a) FROM tabulka_jméno
SELECT *	SELECT * FROM tabulka_jméno  (Poznámka: '*' je zástupný znak za všechny sloupce/atributy dané datové entity !!!)
SELECT DISTINCT	SELECT DISTINCT sloupec_jméno(a) FROM tabulka_jméno  (Poznámka: <b>DISTINCT</b> je klíčové slovo pro unikátní výběr z neunikátní množiny dat v definovaném rozsahu)
SELECT INTO	SELECT * INTO nová_tabulka_jméno [IN externí_databáze] FROM stará_tabulka_jméno  <i>nebo</i>  SELECT sloupec_jméno(a) INTO nová_tabulka_jméno [IN externí_databáze] FROM stará_tabulka_jméno
SELECT TOP	<b>MS SQL Server:</b> SELECT TOP počet_řádků procento_záznamů sloupec_jméno(a) FROM tabulka_jméno  <b>Oracle:</b> SELECT sloupec_jméno(a) FROM tabulka_jméno WHERE ROWNUM <= číslo  <b>MySQL:</b> SELECT sloupec_jméno(a) FROM tabulka_jméno LIMIT číslo
UNION	SELECT sloupec_jméno(a) FROM tabulka_jméno1 UNION SELECT sloupec_jméno(a) FROM tabulka_jméno2
UNION ALL	SELECT sloupec_jméno(a) FROM tabulka_jméno1 UNION ALL SELECT sloupec_jméno(a) FROM tabulka_jméno2
WHERE	SELECT sloupec_jméno(a) FROM tabulka_jméno WHERE sloupec_jméno operátor hodnota

## DDL (Data Definition Language) část:

SQL Příkaz	Syntaxe
ALTER DATABASE	ALTER DATABASE databáze_jméno MODIFY volby  <b>Oracle:</b> <code>ALTER USER databáze_jméno volby (IDENTIFIED BY) heslo</code>
ALTER TABLE	ALTER TABLE tabulka_jméno ADD sloupec_jméno datový_typ  <i>nebo</i>  ALTER TABLE tabulka_jméno DROP SLOUPEC sloupec_jméno
CREATE DATABASE	CREATE DATABASE databáze_jméno  <b>Oracle:</b> <code>CREATE USER databáze_jméno IDENTIFIED BY heslo DEFAULT tablespace users</code>
CREATE TABLE	CREATE TABLE tabulka_jméno ( sloupec_jméno1 datový_typ, sloupec_jméno2 datový_typ, sloupec_jméno2 datový_typ, ... )
CREATE INDEX	CREATE INDEX index_jméno ON tabulka_jméno (sloupec_jméno)  or  CREATE UNIQUE INDEX index_jméno ON tabulka_jméno (sloupec_jméno)
CREATE VIEW	CREATE VIEW pohled_jméno AS SELECT sloupec_jméno(a) FROM tabulka_jméno WHERE podmínka
DROP DATABASE	DROP DATABASE databáze_jméno  <b>Oracle:</b> <code>DROP USER databáze_jméno</code>
DROP INDEX	DROP INDEX tabulka_jméno.index_jméno (SQL Server) DROP INDEX index_jméno ON tabulka_jméno (MS Access) DROP INDEX index_jméno (DB2/Oracle) ALTER TABLE tabulka_jméno DROP INDEX index_jméno (MySQL)
DROP TABLE	DROP TABLE tabulka_jméno

## DML (Data Modification Language) část:

SQL Příkaz	Syntaxe
DELETE	<pre>DELETE FROM tabulka_jméno WHERE libovolný_sloupec=libovolná_hodnota</pre> <p><i>nebo</i></p> <pre>DELETE FROM tabulka_jméno (Poznámka: <b>POZOR vymaže všechny záznamy tabulky !!!</b>)</pre> <p><i>nebo</i></p> <pre>DELETE * FROM tabulka_jméno (Poznámka: <b>POZOR vymaže všechny záznamy tabulky !!!</b>)</pre>
INSERT INTO	<pre>INSERT INTO tabulka_jméno HODNOTAS (hodnota1, hodnota2, hodnota3,...)</pre> <p><i>nebo</i></p> <pre>INSERT INTO tabulka_jméno (sloupec1, sloupec2, sloupec3,...) VALUES (hodnota1, hodnota2, hodnota3,...)</pre>
TRUNCATE TABLE	<pre>TRUNCATE TABLE tabulka_jméno</pre>
UPDATE	<pre>UPDATE tabulka_jméno SET sloupec1=hodnota, sloupec2=hodnota,... WHERE libovolný_sloupec=libovolný_hodnota</pre> <p><i>nebo</i></p> <pre>UPDATE tabulka_jméno SET sloupec1=hodnota, sloupec2=hodnota,... (Poznámka: <b>POZOR aktualizuje záznamy tabulky !!!</b>)</pre>

## Důležité agregační funkce:

Agregační funkce	Popis
AVG()	Vrací průměrnou hodnotu dané (agregované) množiny dat
COUNT()	Vrací počet řádků (agregované) množiny dat
FIRST()	Vrací první hodnotu (agregované) množiny dat
LAST()	Vrací poslední hodnotu dané (agregované) množiny dat
MAX()	Vrací největší hodnotu dané (agregované) množiny dat
MIN()	Vrací nemenší hodnotu dané (agregované) množiny dat
SUM()	Vrací sumu všech hodnot dané (agregované) množiny dat

## Základní divoké znaky:

Wildcard/Divoký znak	Popis
%	Nahrazuje jakékoli znaky v řetězci včetně " s výjimkou hodnoty NULL
_	Nahrazuje jakýkoli jednotlivý znak
[seznam_znaků]	Verifikuje vyjmenované znaky na jediné pozici  <b>Oracle:</b> nepodporuje, nutno nahradit funkcemi INSTR, TRANSLATE, REPLACE apod.
[^seznam_znaků]	Neguje vyjmenované znaky na jediné pozici
nebo	<b>Oracle:</b> nepodporuje, nutno nahradit funkcemi INSTR, TRANSLATE, REPLACE apod.
[!seznam_znaků]	

## Důležité konverzní a substituční funkce:

Funkce	Popis
NVL(str, náhrada), NVL(číslo, náhr.) NVL(date, náhr.)	nahrazuje NULL hodnotu požadovanou náhradou
TO_CHAR(str)	Konvertuje numerickou nebo date hodnotu do stringu
TO_NUMBER(str)	Konvertuje stringovou hodnotu na číslo
ROUND(str, pozic)	Zaokrouhluje numerickou hodnotu dle zadaného počtu pozic
UPPER(str)	Konvertuje string na Upper Case (velké znaky)
LOWER(str)	Konvertuje string na Lower Case (malé znaky)

## Důležité logické operátory:

Funkce	Popis
$A = B$	<b>rovná se</b> , vrací .T. pokud jsou porovnávané hodnoty identické
$A \neq B$ nebo $A <> B$	<b>nerovná se</b> , vrací .T. pokud jsou porovnávané hodnoty rozdílné
$A > B$	.T. pokud je hodnota <b>A větší</b> než hodnota <b>B</b>
$A < B$	.T. pokud je hodnota <b>A menší</b> než hodnota <b>B</b>
$C \text{ BETWEEN } A \text{ AND } B$	.T. pokud je hodnota <b>C</b> v <b>intervalu hodnot</b> <b>A až B</b>  <i>Důležitá poznámka: POZOR závisí na pořadí A a B, a musí být v daném oboru hodnoty menší, jinak se jedná o neexistující interval</i>
$A \text{ LIKE } B$	operátor pro porovnávání hodnot pomocí <b>divokých znaků</b>  <i>Důležitá poznámka: ostatní operátory pracují s divokým znakem jako s běžnou stringovou hodnotou, pouze operátor LIKE respektuje divoké znaky v plné míře</i>
$\text{IN } (A, B, C)$	.T. pokud se hodnota nalézá v seznamu hodnot IN (hodnota1,hodnota2,..)

**Důležitá poznámka: pomocí logických operátorů není možno porovnávat NULL a NOT NULL hodnoty, v případě že podobné porovnání potřebujete provést, je nutno využít konverzní funkce NVL, nebo specializovaného operátoru pro porovnávání hodnot NOT NULL**