

Ultimate SQL Cheat Sheet



SQL KEYWORDS

SQL KEYWORDS

KEYWORDS	DESCRIPTION	KEYWORDS	DESCRIPTION
ADD	<p>Adds a new column to an existing table.</p> <p>Example: Adds a new column named 'email_address' to a table named 'users'.</p> <pre>ALTER TABLE users ADD email address varchar(255);</pre>	DROP COLUMN	<p>Deletes a column from a table.</p> <p>Example: Removes the first_name column from the users table.</p> <pre>ALTER TABLE users DROP COLUMN first_name</pre>
ADD CONSTRAINT	<p>It creates a new constraint on an existing table, which is used to specify rules for any data in the table.</p> <p>Example: Adds a new PRIMARY KEY constraint named 'user' on columns ID and SURNAME.</p> <pre>ALTER TABLE users ADD CONSTRAINT user PRIMARY KEY (ID, SURNAME);</pre>	DROP DATABASE	<p>Deletes the entire database.</p> <p>Example: Deletes a database named 'websitesetup'.</p> <pre>DROP DATABASE websitesetup;</pre>
ALTER TABLE	<p>Adds, deletes or edits columns in a table. It can also be used to add and delete constraints in a table, as per the above.</p> <p>Example: Adds a new boolean column called approved to a table named 'deals'.</p> <pre>ALTER TABLE deals ADD approved boolean;</pre> <p>Example 2: Deletes the approved column from the 'deals' table.</p> <pre>ALTER TABLE deals DROP COLUMN approved;</pre>	DROP DEFAULT	<p>Removes a default value for a column.</p> <p>Example 1(MySQL): Removes the default value from the name column in the 'products' table.</p> <pre>ALTER TABLE products ALTER COLUMN name DROP DEFAULT;</pre>
		DROP TABLE	<p>Deletes a table from a database.</p> <p>Example: Removes the users table.</p> <pre>DROP TABLE users;</pre>
			<p>Checks for the existence of any record within the subquery, returning true if one or more records are returned.</p> <p>Example: Lists any dealerships with a deal</p>

ALTER COLUMN	Changes the data type of a table's column. Example : In the 'users' table, make the column 'Incept_date' into a 'datetime' type. ALTER TABLE users ALTER COLUMN incept_date datetime;	EXISTS	finance percentage less than 10. SELECT dealership_name FROM dealerships WHERE EXISTS (SELECT deal_name FROM deals WHERE dealership_id = deals.dealership_id AND finance_ percentage < 10);
ALL	Returns true if all of the subquery values meet the passed condition. Example : Returns the users with a higher number of tasks than the user with the highest number of tasks in the HR department (id 2). SELECT first_name, surname, tasks_no FROM users WHERE tasks_no > ALL (SELECT tasks FROM user WHERE department_id = 2);	FROM	Specifies which table to select or delete data from. Example : Selects data from the users table. SELECT area_manager FROM area_managers WHERE EXISTS (SELECT ProductName FROM Products WHERE area_manager_id = deals.area_manager_id AND Price < 20);
AND	Used to join separate conditions within a WHERE clause. Example : Returns events located in London, United Kingdom. SELECT * FROM events WHERE host_country='United Kingdom' AND host_city='London';	IN	Used alongside a WHERE clause as a shorthand for multiple OR conditions. So instead of: SELECT * FROM users WHERE country = 'USA' OR country = 'United Kingdom' OR country = 'Russia' OR country = 'Australia'; You can use:- SELECT * FROM users WHERE country IN ('USA', 'United Kingdom', 'Russia', 'Australia');
ANY	Returns true if any of the subquery values meet the given condition. Example : Returns products from the products table which have received orders – stored in the orders table – with a quantity of more than 5. SELECT name FROM products WHERE productId = ANY (SELECT productId FROM orders WHERE Qty >	INSERT INTO	Add new rows to a table. Example : Adds a new vehicle. INSERT INTO cars (make, model, mileage, year) VALUES ('Audi', 'A3', 30000, 2016);
AS	Renames a table or column with an alias value which only exists for the duration of the query. Example : Aliases north_east_user_subscriptions column. SELECT north_east_user_subscriptions AS ne_subs FROM users WHERE ne_subs > 5;	IS NULL	Tests for empty (NULL) values. Example : Returns users that haven't given a contact number. SELECT * FROM users WHERE contact_number IS NULL;
ASC	Used with ORDER BY to return the data in ascending order. Example : Apples, Bananas, Peaches, Raddish.	IS NOT NULL	The reverse of NULL Tests for values that aren't empty / NULL.
BETWEEN	Selects values within the given range. Example 1 : Selects stock with a quantity between 100 and 150. SELECT * FROM stock WHERE quantity BETWEEN 100 AND 150; Example 2 : Selects stock with a quantity NOT between 100 and 150. Alternatively, using the NOT keyword here reverses the logic and selects values outside the given range. SELECT * FROM stock WHERE quantity NOT BETWEEN 100 AND 150;	LIKE	Returns true if the operand value matches a pattern. Example : Returns true if the user's first_name ends with 'son'. SELECT * FROM users WHERE first_name LIKE '%son';
		NOT	Returns true if a record DOESN'T meet the condition. Example : Returns true if the user's first_name doesn't end with 'son'. SELECT * FROM users WHERE first_name NOT LIKE 'son';
CASE	Change query output depending on conditions. Example 1 : Returns users and their subscriptions, along with a new column called activity_levels that makes a judgement based on the number of subscriptions. SELECT first_name, surname, subscriptions CASE WHEN subscriptions > 10 THEN 'Very active'	OR	Used alongside WHERE to include data when either condition is true. Example : Returns users that live in either Sheffield or Manchester. SELECT * FROM users WHERE city = 'Sheffield' OR 'Manchester';
		ORDER BY	Used to sort the result data in ascending (default) or descending order through the use of ASC or DESC keywords. Example : Returns countries in alphabetical order. SELECT * FROM countries ORDER BY name;
			Returns results where the row number meets the passed condition.

	WHEN Quantity BETWEEN 3 AND 10 THEN 'Active' ELSE 'Inactive' END AS activity_levels FROM users;	ROWNUM	Example : Returns the top 10 countries from the countries table. SELECT * FROM countries WHERE ROWNUM C= 10;
CHECK	Adds a constraint that limits the value which can be added to a column. Example 1(MySQL): Makes sure any users added to the users table are 18 or over. CREATE TABLE users first_name varchar(255), age int, CHECK (age=18) ; Example 2(MySQL): Adds a check after the table has already been created. ALTER TABLE users ALTER TABLE users	SELECT	Used to select data from a database, which is then returned in a results set. Example : Selects all columns from all users. SELECT * FROM users; Example 2: Selects the first_name and surname columns from all users.XX SELECT first_name, surname FROM users;
CREATE DATABASE	Creates a new database. Example 1(MySQL): Creates a new database named websiteset. CREATE DATABASE websiteset;	SELECT DISTINCT	Sames as SELECT, except duplicate values are excluded. Example : Creates a backup table using data from the users table. SELECT * INTO usersBackup2020 FROM users;
CREATE TABLE	Creates a new table. Example: Creates a new table called 'users' in the websiteset database. CREATE TABLE users id int, first_name varchar(255), surname varchar(255), address varchar(255), contact number int ;	SELECT INTO	Copies data from one table and inserts it into another. Example: Returns all countries from the users table, removing any duplicate values (which would be highly likely). SELECT DISTINCT country from users;
		SELECT TOP	Allows you to return a set number of records to return from a table. Example: Returns the top 3 cars from the cars table. SELECT TOP 3 * FROM cars;
DEFAULT	Sets a default value for a column; Example 1(MySQL): Creates a new table called Products which has a name column with a default value of 'Placeholder Name' and an available_from column with a default value of today's date. CREATE TABLE products id int, name varchar(255) DEFAULT 'Placeholder Name', available from date DEFAULT GETDATE() ; Example 2(MySQL): The same as above, but editing an existing table. ALTER TABLE products ALTER name SET DEFAULT 'Placeholder Name' ALTER available_from SET DEFAULT GETDATE();	SET	Used alongside UPDATE to update existing data in a table. Example: Updates the value and quantity values for an order with an id of 642 in the orders table. UPDATE orders SET value = 19.49, quantity = 2 WHERE id = 642;
		SOME	Identical to ANY
		TOP	Used alongside SELECT to return a set number of records from a table. Example: Returns the top 5 users from the users table. SELECT TOP 5 * FROM users;
		TRUNCATE TABLE	Similar to DROP, but instead of deleting the table and its data, this deletes only the data. Example: Empties the sessions table, but leaves the table itself intact. TRUNCATE TABLE sessions;
DELETE	Delete data from a table. Example: Removes a user with a user_id of 674. DELETE FROM users WHERE user_id = 674;	UNION	Combines the results from 2 or more SELECT statements and returns only distinct values. Example: Returns the cities from the events and subscribers tables. SELECT city FROM events UNION SELECT city from subscribers;
DESC	Used with ORDER BY to return the data in descending order. Example: Raddish, Peaches, Bananas, Apples.	UNION ALL	The same as UNION but includes duplicate values.
			This constraint ensures all values in a column are unique. Example 1 (MySQL): Adds a unique constraint to the id column when creating a new users table. CREATE TABLE users id int NOT NULL,