

عنوان ها	دستور	کاربرد	مثال	توضیحات
SELECT	SELECT	انتخاب کردن داده ها	SELECT coulmn1 FROM table; SELECT * FROM table;	یک ستون همه ستون ها
	DISTINCT	مقادیر تکراری حذف می شود	SELECT DISTINCT coulmn1 FROM table;	مقادیر یونیک یک ستون
	COUNT	تعداد مقادیر	SELECT COUNT(coulmn1) FROM table;	تعداد مقادیر های یک ستون
	OPERATORS	اگر TYPE مقادیر از یک نوع باشند و در یک جدول قرار داشته باشند	SELECT coulmn1, (coulmn2 - coulmn3) FROM table;	ستون دوم منهای ستون سوم
WHERE	WHERE	قرار دادن شرط روی کونری ها	SELECT coulmn1 FROM table WHERE coulmn1 > 10;	ستون هایی که مقادیر بزرگتر از ۱۰ دارند
	AND	و	SELECT coulmn1 FROM table WHERE coulmn1 > 10 AND coulmn1 < 3;	ستون هایی که مقادیر بزرگتر از ۱۰ دارند و کوچکتر از ۳
	OR	یا	SELECT coulmn1 FROM table WHERE coulmn1 = 10 OR coulmn1 = 3;	ستون هایی که مقادیر برابر ۱۰ یا ۳
	NOT	نه	SELECT coulmn1 FROM table WHERE NOT coulmn1 < 3;	ستون های که مقادیر آنها کمتر از ۳ نباشد
	BETWEEN	بین	SELECT coulmn1 FROM table WHERE coulmn1 BETWEEN 10 AND 3;	ستون های که مقادیر آنها بین ۱۰ و ۳ باشد
	IN	در	SELECT coulmn1 FROM table WHERE country IN (2, 3, 4, 5);	ستون هایی که مقادیرشان در لیست (۲ و ۳ و ۴ و ۵) باشد
ORDER BY	ASC	مرتب شده به ترتیب حروف الفبا از بالا به پایین	SELECT coulmn1 FROM table ORDER BY ASC;	ستون ها بر اساس حروف الفبا از بالا به پایین مرتب شده اند
	DESC	مرتب شده به ترتیب حروف الفبا از پایین به بالا	SELECT coulmn1 FROM table ORDER BY DESC;	ستون ها بر اساس حروف الفبا از پایین به بالا مرتب شده اند
	MAX	بیشینه	SELECT MAX(coulmn1) FROM table;	بزرگترین مقدار ان ستون
	MIN	کمینه	SELECT MIN(coulmn1) FROM table;	کوچکترین مقدار ان ستون
	AVG	میانگین	SELECT AVG(coulmn1) FROM table;	میانگین مقادیر ان ستون
	SUM	مجموع	SELECT SUM(coulmn1) FROM table;	مجموع مقادیر ان ستون
LIKE	%	هیچی یا هر تعداد	SELECT coulmn1 FROM table WHERE coulmn1 LIKE 'A%';	مقادیری که حرف اولشان A باشد و بعد از ان هیچ یا هر تعداد حرف
	—	یک	SELECT coulmn1 FROM table WHERE coulmn1 LIKE 'A_f%';	مقادیری که حرف اولشان A باشد حرف دومشان هر چی بود حرف سومشان f باشد و بعد از ان هیچ یا هر تعداد حرف
AS LIMIT	AS	اسم مستعار	SELECT coulmn1 AS c FROM table;	ستون انتخاب شده را با نام C میثناسد
	LIMIT	محدود کردن تعداد سطر های یک جدول	SELECT coulmn1 FROM table LIMIT 30;	۳۰ مقدار اول ان ستون را برمیگرداند

JOIN	INNER JOIN	چندین ستون را بهم میچسباند	SELECT coulmn1, coulmn2, coulmn3 FROM table1 INNER JOIN table2 ON table1.coulmn(m) = table2.coulmn(m) INNER JOIN table3 ON table2.coulmn(n) = table3.coulmn(n);	ستون ۱ از جدول ۱ - ستون ۲ از جدول ۲ و ستون ۳ از جدول ۳ بهم متصل شده اند زیرا ستون m از جدول ۱ و جدول ۲ و ستون n از جدول ۲ و جدول ۳ با هم مساوی هستند
	LEFT JOIN	چندین ستون را بهم میچسباند برای مقادیری که در ستون سمت چپ وجود دارند ولی مقدار متناظر آنها در ستون سمت راست نیست NULL برمیگرداند	SELECT coulmn1, coulmn2 FROM table1 LEFT JOIN table2 ON table1.coulmn(m) = table2.coulmn(m);	مقادیری از ستون سمت راست که متناظر با ستون سمت چپ وجود ندارد را NULL قرار میدهد.
	RIGHT JOIN	چندین ستون را بهم میچسباند برای مقادیری که در ستون سمت راست وجود دارند ولی مقدار متناظر آنها در ستون سمت چپ نیست NULL برمیگرداند	SELECT coulmn1, coulmn2 FROM table1 RIGHT JOIN table2 ON table1.coulmn(m) = table2.coulmn(m);	مقادیری از ستون سمت چپ که متناظر با ستون سمت راست وجود ندارد را NULL قرار میدهد.
	FULL JOIN	تمامی مقادیر ستون ۱ و ستون ۲ را برمیگرداند حتی اگر مقدار متناظری نداشته باشند.	SELECT coulmn1, coulmn2 FROM table1 FULL JOIN table2 ON table1.coulmn(m) = table2.coulmn(m);	تمامی مقادیر ستون ۱ و ستون ۲ را بر می گرداند حتی اگر مقدار متناظری نداشته باشند
	SELF JOIN	برای مقایسه مقادیر یک ستون با خودش	SELECT c1.coulmn1, c2.coulmn1 FROM table1 c1, table1 c2 WHERE c1.coulmn1 > c2.coulmn1;	برای مقایسه مقادیر یک ستون با خودش
GROUP BY	GROUP BY	وقتی بخواهیم بر اساس یک ستون نتایج را دسته بندی کنیم	SELECT coulmn1, COUNT(*) FROM table1 GROUP BY coulmn1;	تعداد تکرار هر مقدار از ستون اول را در ستون دوم مینویسد در ستون اول مقادیر تکراری را فقط یکبار مینویسد
HAVING	HAVING	وقتی بخواهیم بعد از گروپ بای شرط بگذاریم نمیتوانیم از نام مستعار استفاده کنیم	SELECT table1.coulmn1, SUM(table2.coulmn2*table3.coulmn3) AS total FROM table1.coulmn1 INNER JOIN table2 ON table2.coulmn(m) = table1.coulmn(m) INNER JOIN table3 ON table3.coulmn(n) = table2.coulmn(n) GROUP BY coulmn1 HAVING SUM(coulmn2*coulmn3) > 2000 ORDER BY table1.coulmn1 ASC;	مجموع حاصل ضرب ستون ۲ و ستون ۳ برای هر آیتم از ستون ۱ را دسته بندی میکند با نام مستعار توتال و برای هر آیتم که ستون توتال آن بزرگ بزرگ تر از ۲۰۰۰ باشد را بر میگرداند و بر اساس ستون ۱ به ترتیب حروف الفبا مرتب میکند
UNION	UNION	دو SELECT را با هم ادغام میکند مقادیر تکراری حذف میشود	SELECT coulmn1 FROM table1 UNION SELECT coulmn1 FROM table2;	ستون ۱ از جدول ۱ را با ستون ۱ از جدول ۲ با حذف آیتم های تکراری ادغام میکند
	UNION ALL	دو SELECT را با هم ادغام میکند مقادیر تکراری حذف نمیشوند	SELECT coulmn1 FROM table1 UNION ALL SELECT coulmn1 FROM table2;	ستون ۱ از جدول ۱ را با ستون ۱ از جدول ۲ با ادغام میکند
INSERT INTO	INSERT INTO	به انتهای ستون های این جدول مقادیر گفته شده را اضافه و سایر ستون ها را NULL قرار میدهد	INSERT INTO table(coulmn1, coulmn2, coulmn3) VALUES ("ALI", 22, "IRAN");	به انتهای ستون های این جدول مقادیر گفته شده را اضافه میکند

UPDATE	UPDATE	یک ردیف override میکند به مقادیر جدید اگر شرط را ننویسیم کل جدول را از بین میبرد	UPDATE table SET coulumn1="ali", coulumn2="hajiali" WHERE coulumn1="reza";	مقدار علی و حاجی علی را جایگزین مقدار رضا و نام خانوادگی رضا میکند	
DELETE	DELETE	یک ردیف را حذف میکند اگر شرط را نگذاریم کل ردیف ها را حذف میکند	DELETE FROM table WHERE coulumn1=10;	ردیفی که ستون ۱ برابر ۱۰ باشد را حذف میکند	
SELECT INTO	SELECT INTO	یک جدول جدید میسازد و از داده هایی که تعیین میکنیم یک آپ میگیرد	SELECT coulumn1, coulumn2 INTO new_table FROM old_table WHERE coulumn1 IN ("iran", "spain", "italia");	از جدول قدیم ستون ۱ و ستون ۲ را داخل جدول جدید میریزد به شرطی که ستون ۱ یکی از ("ایران", "اسپانیا", "ایتالیا") باشد	
	INSERT INTO SELECT	از داده هایی که تعیین میکنیم یک آپ میگیرد و داخل یکی دیگر از جدول های (موجود) میریزد	INSERT INTO table2 SELECT coulumn1, coulumn2 FROM table1 WHERE coulumn1 > 11 ;	از جدول ۱ ستون ۱ و ستون ۲ را داخل جدول ۲ میریزد به شرطی که ستون ۱ بزرگتر از ۱۱ باشد	
CREATE, DROP	CREATE	ایجاد دیتابیس	CREATE DATABASE db_name;	ایجاد دیتابیس	
	DROP	حذف دیتابیس	DROP DATABASE db_name;	حذف دیتابیس	
	DROP TABLE	حذف جدول	DROP TABLE table_name;	حذف جدول	
CREATE TABLE	CREATE TABLE	ساخت جدول	CREATE TABLE table_name (coulumn1 DATA TYPE, coulumn2 DATA TYPE);	یک جدول شامل دو ستون با نوع داده که در لینک سلول بغلی توضیح داده شده است ساخته ایم	DATA TYPE
ALTER	RENAME COULMN	تغییر نام ستون ها	ALTER TABLE table RENAME coulumn_oldname TO coulumn_newname;	تغییر نام ستون ها	
	RENAME TABLE	تغییر نام جدول	ALTER TABLE table RENAME TO table_newname;	نام جدول تغییر کرد	
	CREATE COULMN	افزودن ستون به جدول	ALTER TABLE table ADD coulumn_name DATE TYPE;	یک ستون به جدول اضافه کردیم	
	DROP COULMN	حذف یک ستون	ALTER TABLE table DROP COULMN coulumn_name;	ستون حذف شد	
	NEW DATA TYPE	تغییر نوع دیتا	ALTER TABLE table ALTER COULMN coulumn_name SET DATA TYPE new_data_type;	نوع دیتا تغییر کرد	
INDEX	INDEX	ایندکس کردن دیتابیس	CREATE INDEX index_name ON table_name (coulumn_name);	دیتابیس ایندکس شد	
	UNIQUE INDEX	ایندکس کردن یونیک دیتابیس	CREATE UNIQUE INDEX index_name ON table_name (coulumn_name);	دیتابیس بصورت یونیک ایندکس شد	
	DROP INDEX	حذف ایندکس	DROP INDEX index_name;	ایندکس حذف شد	
	NOT NULL	خالی نباشه	CREATE TABLE table_name(coulumn1 DATA TYPE NOT NULL);	یک ستون در یک جدول جدید ساختیم که نمیتونه خالی باشه	

CONSTRAINT	UNIQUE	منحصر به فرد است	CREATE TABLE table_name(coulmn1 DATA TYPE UNIQUE);	منحصر به فرد است
	PRIMARY KEY	کلید داخلی	CREATE TABLE table_name(coulmn1 DATA TYPE PRIMARY KEY);	کلید داخلی
	FOREIGN KEY	کلید خارجی	CREATE TABLE table_name(coulmn1 DATA TYPE NOT NULL PRIMARY KEY, coulmn2 DATA TYPE, FOREIGN KEY (coulmn(m)) REFERENCES other_table(coulmn(m)));	کلید خارجی
	DEFAULT	مقدار پیش فرض بر عکس NOT NULL	CREATE TABLE table_name(coulmn1 DATA TYPE NOT NULL PRIMARY KEY, coulmn2 DATA TYPE DEFAULT value);	مقدار پیش فرض
	CHECK	چک کردن مقادیری که قرار است در هر ستون اضافه شوند	CREATE TABLE table_name(coulmn1 DATA TYPE , coulmn2 DATA TYPE CONSTRAINT check_name CHECK (coulmn2 > 22)); or CREATE TABLE table_name(coulmn1 DATA TYPE , coulmn2 DATA TYPE CONSTRAINT check_name CHECK (coulmn2 > 22 AND LENGTH (coulmn1) > 3));	روش اول : چک کردن برای یک ستون برای چند ستون چند بار نوشته میشود چک میکند مقدار ورودی در ستون ۲ بزرگ تر از ۲۲ باشد در غیر اینصورت هیچ مقدار نمیگیرد . روش دوم : چک کردن چندین ستون چک میکند که مقدار ورودی در ستون ۲ بزرگ تر از ۲۲ باشد و ستون ۱ بزرگتر از ۳ باشد در غیر اینصورت هیچ مقداری نمیگیرد
	DROP CONSTRAINT	حذف یک check	ALTER TABLE table_name DROP CONSTRAINT check_name	check حذف شد