SQL Cheat Sheet

Statement	Syntax	Quick Description
SELECT	**SELECT** [**DISTINCT**] * | LIST OF COLUMNS, FUNCTIONS, CONSTANTS **FROM** LIST OF TABLES OR VIEWS [**WHERE** CONDITION(S) [AND|OR|NOT]] [**ORDER BY** ORDERING COLUMN(S) [ASC | DESC]] [**GROUP BY** GROUPING COLUMN(S)] [**HAVING** CONDITION(S)]	Query a set of records. ***Equivalent to*** $\Pi_{A1, A2... An} (\sigma_P (r_1 x\ r_2 x \dots r_n))$ ***and returns a relation.*** **SELECT** * **FROM** r1, r2 is Cartesian Product Functions allowed: SUM, AVG, MIN, MAX, COUNT **DISTINCT** removes duplicates **NOTE:** ALL QUOTES AROUND STRINGS ARE SINGLE QUOTES! To reference a column in a specific relation, use TABLE.COLUMN_NAME
WHERE	**...WHERE** COLUMN OPERATOR VALUE Set comparisons: **...WHERE** COLUMN OPERATOR **ALL | SOME** **...WHERE [NOT] EXISTS** RELATION **...WHERE [NOT] UNIQUE** RELATION	Operators: =, <> (not equal), >, >=, <, <=, BETWEEN..AND (inclusive range), LIKE (pattern search – use % for wildcards, _ for single character), IN (find value in a set) Value: string in single quotes, numerical value, another attribute, or **IS NULL**
ORDER BY	**...ORDER BY** COLUMN(S) [ASC | DESC]	Sorts results by given columns either ascending or descending
GROUP BY	**...GROUP BY** COLUMN(S)	Used with aggregate functions (like the first part before the huge 'g') to group functions
HAVING	**...HAVING** CONDITION(S)	
UNION INTERSECT SET DIFFERENCE	QUERY **UNION** QUERY QUERY **INNER JOIN** QUERY **SELECT** * from P **LEFT JOIN** Q on P.id = Q.id where Q.id is NULL;	Union two relations Find the intersection of two relations Perform set difference on two relations
DELETE	**DELETE FROM** TABLE NAME [**WHERE** CONDITION(S)]	Delete all the records in a table (or those which match the condition(s))

INSERT	**INSERT INTO** TABLE NAME
	[(COLUMN LIST)]
	VALUES (VALUE LIST)
		Insert records into a table

UPDATE	**UPDATE** TABLE NAME
	SET COLUMN NAME = VALUE
	[**WHERE** CONDITION]
		Modify a column/field value in a table, (or in records in the table which match the condition(s)

CREATE TABLE	**CREATE TABLE** TABLE_NAME (COLUMN_NAME DATA_TYPE [(SIZE)] COLUMN_CONSTRAINT, [, other column definitions,...] [, primary key constraint])	Create a new table
CREATE VIEW	**CREATE VIEW** VIEW_NAME **AS** QUERY_NAME	Create a new view
WITH	**WITH** NEW_TABLE(NEW_COLUMNS) **AS** QUERY	Creates a local view only used in following queries
ALTER	**ALTER TABLE** TABLE_NAME **ADD** | **DROP** | **MODIFY** (COLUMN_NAME DATA_TYPE [(SIZE)] COLUMN_CONSTRAINT, [, other column definitions,...])	Alter table columns, remove/modify/add columns from (or to) a table.
DROP TABLE	**DROP TABLE** TABLE_NAME	Delete an entire table
AS	...COLUMN_NAME **AS** COLUMN_NAME	Renames; mainly used to create derived attributes
CONSTRAINT	**CONSTRAINT** CONSTRAINT_NAME {PRIMARY KEY | UNIQUE | NOT NULL | **REFERENCES** FOREIGN_TABLE [(FIELD_LIST)]}	Create a table constraint, (with references from another table)

Data types in SQL: char(n), varchar(n), int, smallint, numeric(precision, after-decimal), real, float, not null, date, time, timestamp