

Admin Functions	Command Line	Official PostgreSQL 8.3 Documentation URL: <a href="http://www.postgresql.org/docs/8.3/static/">http://www.postgresql.org/docs/8.3/static/</a> We cover only a subset of what we feel are the most useful constructs that we could squash in a single cheatsheet page
<code>COPY .. FROM ..</code> <code>COPY .. TO ..</code> <code>current_setting</code> <code>pg_cancel_backend</code> <code>pg_column_size</code> <code>pg_database_size</code> <code>pg_relation_size</code> <code>pg_size_pretty</code> <code>pg_tablespace_size</code> <code>pg_total_relation_size</code> <code>set_config</code> <code>vacuum analyze verbose</code> <code>vacuum full</code>	<code>pg_dump</code> <code>pg_dumpall</code> <code>pg_restore</code> <code>psql</code>	<b>commonly used</b> <small>*New in this release</small>
<b>Common Functions</b>	<b>JOIN Types</b>	<b>DATA TYPES</b>
<code>cast, ::</code> <code>coalesce</code> <code>generate_series</code> <code>greatest</code> <code>least</code> <code>nullif</code> <code>random</code>	<b>SQL Keywords</b> BETWEEN .. AND CASE WHEN .. END DELETE FROM DISTINCT DISTINCT ON EXISTS FROM GROUP BY HAVING LIKE LIKE LN(..) NOT NOT IN(..) NULLS FIRST <sup>1</sup> NULLS LAST <sup>1</sup> ORDER BY SELECT SET SIMILAR TO TRUNCATE TABLE UPDATE USING WHERE	<code>numeric(length,precision)</code> <code>oid</code> <code>serial - serial4</code> <code>bigserial - serial8</code> <code>text</code> <code>time without timezone - time</code> <code>time with timezone - timez</code> <code>timestamp without timezone - timestamptz</code> <code>timestamp with timezone - timestamptz</code> <code>xml 1</code>
<b>Sequence (Serial) Functions</b>		
<code>currval</code> <code>lastval</code> <code>nextval</code>		
<b>String Functions</b>		
<code>  </code> <code>ascii</code> <code>chr</code> <code>initcap</code> <code>length</code> <code>lower</code> <code>lpad</code> <code>ltrim</code> <code>md5</code> <code>position</code> <code>quote_ident</code> <code>quote_literal</code> <code>regexp_matches</code> <code>regexp_replace</code> <code>regexp_split_to_array</code> <code>regexp_split_to_table</code> <code>repeat</code> <code>replace</code> <code>rpad</code> <code>rtrim</code> <code>split_part</code> <code>strpos</code> <code>substr</code> <code>trim</code> <code>upper</code>		
<b>Database Globals</b>		
<code>current_date</code> <code>current_time</code> <code>current_timestamp</code> <code>current_user</code> <code>localtime</code>		
<b>Date Functions</b>		
<code>age</code> <code>date_part(text, timestamp)</code> <code>century</code> <code>day</code> <code>decade</code> <code>dow</code> <code>day</code> <code>epoch</code> <code>hour</code> <code>month</code> <code>quarter</code> <code>second</code> <code>week</code> <code>year</code> <code>date_trunc</code> <code>extract</code> <code>interval</code> <code>to_char</code> <code>to_date</code> <code>to_timestamp</code>	<b>Aggregates</b> avg bit_and bit_or boolean_and boolean_or count count(DISTINCT) every max min stddev stddev_pop (a bunch more) sum sum(DISTINCT) variance xml_agg <sup>1</sup>	<b>ADMIN EXAMPLES</b> <pre>select pg_size_pretty(pg_tablespace_size('pg_default')) as tssize,        pg_size_pretty(pg_database_size('somedb')) as dbsize,        pg_size_pretty(pg_relation_size('someschema.sometable')) as tbsize;</pre> <small>--Example importing data to table sometable --from tab delimited where NULLs appear as NULL COPY sometable FROM '/path/to/textfile.txt' USING DELIMITERS '\t' WITH NULL AS 'NULL';</small>  <small>--Example exporting a query to a comma separated (CSV) called textile.csv --setting NULLs to text NULL COPY (SELECT * FROM sometable WHERE somevalue LIKE '%') TO '/path/to/textfile.csv' WITH NULL AS 'NULL' CSV HEADER QUOTE AS '';</small>  <code>vacuum analyze verbose;</code> <code>vacuum sometable;</code> <code>vacuum full;</code>  <small>--Kills all active queries in selected db and list out process id --and username of process and if all successful SELECT procpid, username, pg_cancel_backend(procpid) FROM pg_stat_activity WHERE datname = 'somedb';</small>
<b>Date Predicates</b>		
<code>overlaps</code>		
<b>Array Constructs</b>		
<code>ANY(array)</code> <code>ARRAY[4,5,6,...]</code> <code>ARRAY()</code> <code>array_append</code> <code>array_cat</code> <code>array_dims</code> <code>array_lower</code> <code>array_prepend</code> <code>array_to_string</code> <code>array_upper</code> <code>SOME(array)</code> <code>string_to_array</code>	<b>Enums</b> <sup>1</sup>	
<b>Array Operators</b>		
<code>=</code> <code>&lt;</code> <code>&gt;</code> <code>&lt;=</code> <code>&gt;=</code> <code>  </code>	<code>enum_cmp</code> <code>enum_first</code> <code>enum_larger</code> <code>enum_last</code> <code>enum_range</code> <code>enum_smaller</code>	
<b>Math Operators</b>		
<code>%</code> <code>^</code> <code>/</code> <code>  /</code> <code>! </code> <code>@</code> <code>&amp;</code> <code> </code> <code>#</code> <code>~</code> <code>&lt;&lt;</code> <code>&gt;&gt;</code>	<b>Languages</b> <code>c</code> <code>pljava</code> <code>plpgsql</code> <code>plperl(u)</code> <code>plphp</code> <code>plpython</code> <code>plr</code> <code>plruby</code> <code>plsh</code> <code>pltcl</code> <code>sql</code>	
<b>Math Functions</b>		
This is a subset <code>abs</code> <code>ceil</code> <code>ceiling</code> <code>degrees</code> <code>exp</code> <code>floor</code> <code>log</code> <code>ln</code> <code>mod</code> <code>pi</code> <code>power</code> <code>radians</code> <code>random</code> <code>sqrt</code> <code>trunc</code>	<b>Key Information schema Views</b> <code>columns</code> <code>sequences</code> <code>tables</code> <code>views</code>	
<b>Trig Functions</b>		
<code>acos</code> <code>asin</code> <code>atan</code> <code>atan2</code> <code>cos</code> <code>cot</code> <code>sin</code> <code>tan</code>	<b>Key pg_catalog Tables/Views</b> <code>pg_class</code> <code>pg_rules</code> <code>pg_settings</code> <code>pg_stat_activity</code> <code>pg_stat_database</code> <code>pg_tablespace</code>	
		<b>JOIN EXAMPLES</b> <pre>SELECT o.order_id, o.order_date, o.approved_date,        COUNT(i.item_id) As nlineitems,        SUM(i.unit_price*i.num_units) As total FROM orders o INNER JOIN orderitems i ON o.order_id = i.order_id GROUP BY o.order_id, o.order_date, o.approved_date HAVING SUM(i.unit_price*i.num_units) &gt; 200 ORDER BY o.approved_date NULLS FIRST;</pre> <pre>SELECT 'x' As bucket, o.order_id, o.order_date,        COUNT(i.item_id) As nlineitems,        SUM(i.unit_price*i.num_units) As total FROM orders o INNER JOIN orderitems i ON o.order_id = i.order_id GROUP BY o.order_id, o.order_date UNION ALL SELECT 'y' as bucket, o.order_id, o.order_date,        COUNT(i.item_id) As nlineitems,        SUM(i.unit_price*i.num_units) As total FROM orders o INNER JOIN orderitems i ON o.order_id = i.order_id GROUP BY o.order_id, o.order_date ORDER BY 1,3,2;</pre>
		<b>DDL EXAMPLES</b> <pre>CREATE DATABASE somedb WITH OWNER = someogin ENCODING = 'WIN1252';</pre> <pre>CREATE TABLE orders(   order_id serial NOT NULL,   order_addeddt timestamp without time zone,   order_rating rating,   CONSTRAINT pk_orders_order_id PRIMARY KEY (order_id) ) WITH (OIDS=FALSE);</pre> <pre>CREATE TYPE rating AS ENUM('none', 'bronze', 'silver',      'gold', 'platinum');</pre> <pre>CREATE AGGREGATE sum(text) (   SFUNC=txtcat,   STYPE=text );</pre> <pre>CREATE OR REPLACE FUNCTION cp_test(somearg integer) RETURNS SETOF sometable AS \$\$SELECT * FROM sometable where msg_id = \$1:\$2 LANGUAGE 'sql' STABLE;</pre>
		<b>UPDATE/INSERT/DELETE EXAMPLES</b> <pre>UPDATE sometable SET calccount = s.thecount FROM (SELECT COUNT(someothertable.someid) as thecount,       someothertable.someid       FROM someothertable       GROUP BY someothertable.someid) s WHERE sometable.someid = s.someid;</pre> <small>--Pre 8.1+ only supports single values inserts</small> <pre>INSERT INTO orders(order_addeddt, order_rating) VALUES ('2007-10-01 20:40', 'gold'), ('2007-09-01 11:00 AM', 'silver'), ('2007-09-02 10:00 PM', 'none'), ('2007-10-10 PM', 'bronze');</pre> <pre>DELETE FROM sometable WHERE somevalue = 'something';</pre> <small>--This is a fast delete that deletes everything in a table so be cautious. --Also only works on tables not referenced in foreign key constraints TRUNCATE TABLE sometable;</small>
		<b>MISCELLANEOUS EXAMPLES</b> <small>--Enum range query using enum defined above - returns all orders in (bronze, silver, gold) --Sorts in order bronze, silver, gold. Keep in mind if you reverse gold and bronze you get nothing</small> <pre>SELECT * FROM orders WHERE order_rating BETWEEN 'bronze' AND 'gold' ORDER BY order_rating;</pre> <pre>SELECT monthperiod, * array_to_string(ARRAY(SELECT (d + 1)::varchar(20) FROM generate_series(0,30) d WHERE monthperiod.start_date + (d    ' day')::interval BETWEEN monthperiod.start_date AND monthperiod.end_date), ',') as thedays FROM (SELECT (n + 1) As mnum, trim(to_char(date '2007-01-01' + (n    ' month')::interval, 'Mon')) As short_mname, trim(to_char(date '2007-01-01' + (n    ' month')::interval, 'Month')) As long_mname, date '2007-01-01' + (n    ' month')::interval As start_date, date '2007-01-01' + ((n + 1)    ' month')::interval + - '1 day'::interval As end_date FROM generate_series(0,11) n) As monthperiod;</pre> <code>EXPLAIN ANALYZE SELECT * FROM sometable;</code>
		<b>COMMAND LINE EXAMPLES</b> <small>These are located in bin folder of PostgreSQL To get more info about each do a -help e.g. psql -help</small> <pre>pg_dump -i -h someserver -p 5432 -U someuser -F c -b -v -f \"somepath/somedb.backup\" somedb pg_dumpall -i -h someserver -p 5432 -U someuser -c -o -f \"somepath/alldbs.sql\" pg_restore -i -h someserver -p 5432 -U someuser -d somedb -l \"somepath/somedb.backup\" psql -h someserver -p 5432 -U someuser -d somedb -f \"somepath/somefile.tsql\" psql -h someserver -p 5432 -U someuser -d somedb -c \"CREATE TABLE sometable(st_id serial, st_name varchar(25));\" psql -h someserver -p 5432 -U someuser -d somedb -P \"t\" -c \"SELECT query_to_xml('select * from sometable', false, false, 'sometable')\" -o 'outputfile.xml';</pre> <small>*New XML feature - output query as xml *T only output rows</small>
		<a href="http://www.postgresql.org">http://www.postgresql.org</a>