**SELECT** column\_name(s) FROM table\_name WHERE condition **AND|OR** condition

ALTER TABLE table\_name **ADD** column\_name datatype **REFERENCES** Filming(id)

ALTER TABLE Movie ALTER COLUMN test TYPE int USING test::integer;

ALTER TABLE Movie RENAME COLUMN test TO test2;

ALTER TABLE Movie DROP COLUMN test;

ALTER TABLE Movie **RENAME** TO Movie2;

**DROP** COLUMN column\_name

SELECT column\_name **AS** column\_alias FROM table\_name; SELECT column\_name FROM table\_name AS table\_alias

SELECT column\_name(s) FROM table\_name WHERE column\_name **BETWEEN** value1 AND value2(包括前后)

**CREATE** **DATABASE** database\_name

CREATE **TABLE** table\_name(column\_name1 data\_type, (id SERIAL PRIMARY KEY, title VARCHAR, tstart DATE CHECK (tstart > '1895-01-01'), tend DATE, CHECK (tstart < tend));

CREATE **INDEX** index\_name ON table\_name (column\_name) (Duplicate values are allowed)

CREATE **INDEX** movie\_tags ON movies **USING** gin (tags);

CREATE **UNIQUE INDEX** index\_name ON table\_name (column\_name) (Duplicate values are not allowed)

CREATE **VIEW** view\_name AS SELECT column\_name(s) FROM table\_name WHERE condition

**DELETE** FROM table\_name WHERE some\_column=some\_value

DELETE FROM table\_name (Note: Deletes the entire table!!); DELETE \* FROM table\_name (Note: Deletes the entire table!!)

**DROP DATABASE** database\_name

**DROP INDEX** table\_name.index\_name (SQL Server); DROP INDEX index\_name ON table\_name (MS Access); DROP INDEX index\_name (DB2/Oracle) ; ALTER TABLE table\_name DROP INDEX index\_name (MySQL)

**DROP TABLE** table\_name

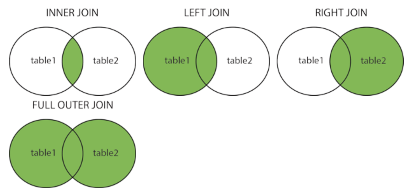
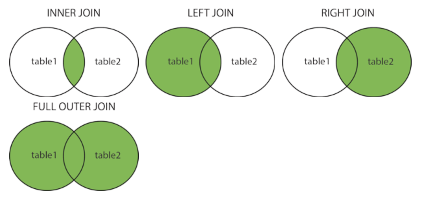
IF **EXISTS** (SELECT \* FROM table\_name WHERE id = ?) BEGIN --do what needs to be done if exists END ELSE BEGIN --do what needs to be done if not END

SELECT column\_name, aggregate\_function(column\_name) FROM table\_name WHERE column\_name operator value **GROUP BY** column\_name **HAVING** aggregate\_function(column\_name) operator value(常与COUNT(), MAX(), MIN(), SUM(), AVG()一起用)

SELECT column\_name(s) FROM table\_name WHERE column\_name **IN** (value1,value2,..)

**INSERT INTO** table\_name VALUES (v1, v2, v3,....); INSERT INTO table\_name (column1, column2, column3,...) VALUES (v1, v2, v3,....)

SELECT column\_name(s) FROM table\_name1 **INNER JOIN** table\_name2 ON table\_name1.column\_name = table\_name2.column\_name

SELECT column\_name(s) FROM table\_name1 **LEFT JOIN** table\_name2 ON …

SELECT column\_name(s) FROM table\_name1 **RIGHT JOIN** table\_name2 ON …

SELECT column\_name(s) FROM table\_name1 **FULL JOIN** table\_name2 ON …

SELECT column\_name(s) FROM table\_name WHERE column\_name **LIKE** pattern **%** :zero, one, or multiple characters, **(\_)** one single character

SELECT column\_name(s) FROM table\_name **ORDER BY** column\_name **[ASC|DESC] LIMIT 5**

**SELECT** **DISTINCT** column\_name(s) FROM table\_name (return only distinct (different) values)

**SELECT \* INTO** new\_table\_name [IN externaldatabase] FROM old\_table\_name (copies data from one table into a new table)

SELECT column\_name(s) INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name

SELECT **TOP** number|percent column\_name(s) FROM table\_name

**TRUNCATE** TABLE table\_name

SELECT column\_name(s) FROM table\_name1 **UNION** SELECT column\_name(s) FROM table\_name2

SELECT column\_name(s) FROM table\_name1 **UNION ALL** SELECT column\_name(s) FROM table\_name2 (allow duplicate values)

**UPDATE** table\_name SET column1=value, column2=value,... WHERE some\_column=some\_value

Comments, with -- or /\* ... \*/

**\l** to list databases **\c database** change database \**dt** list tables **\d table** show details about a table

PostgreSQL types : **BOOLEAN** , **INT** for integers (4-byte) **SERIAL** for an auto-incrementing identifier (4-byte), or AUTO INCREMENT with MySQL **REAL** for floating-point numbers (4-byte) **NUMERIC** for high-precision numbers (1000 digits) • **TEXT or VARCHAR**: text • VARCHAR(42): text of length at most 42 • **BYTEA** or **BLOB** for binary strings • **TIMESTAMP** for date and time (can be WITH TIME ZONE), DATE, etc