

Γλώσσα SQL ακαδ. έτος 2021-22

Διδάσκων: καθ. Γιάννης Θεοδωρίδης

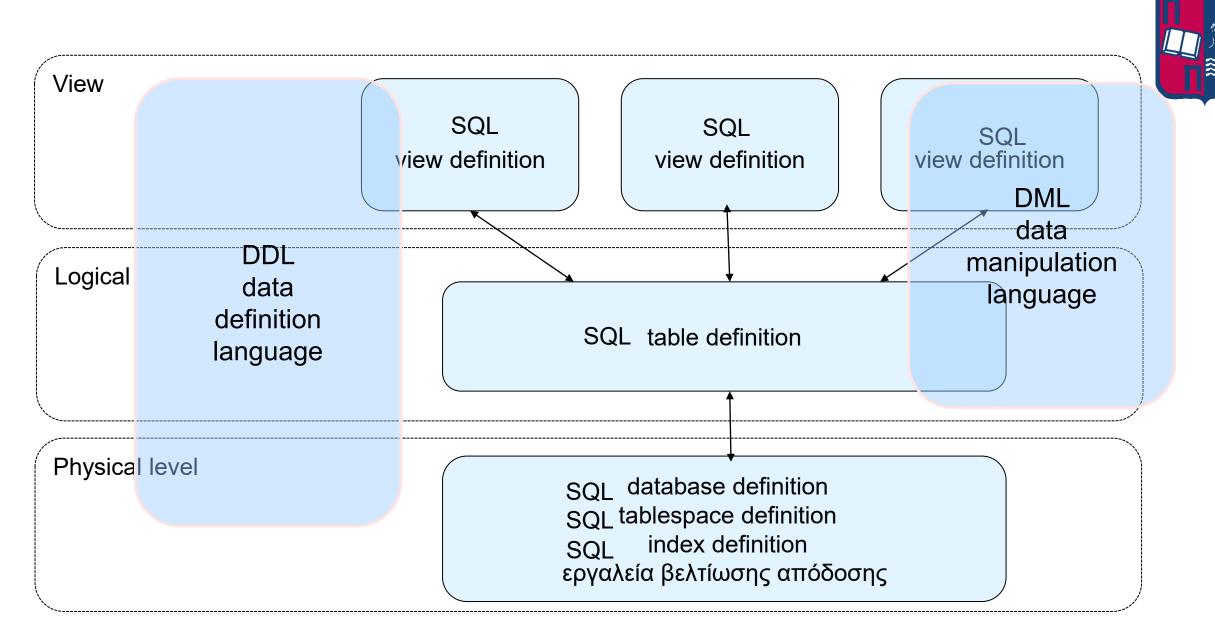
+Διδάσκουσα: Δρ. Μαυροπόδη Ρόζα

Εργαστηριακοί βοηθοί: Γιάννης Κοντούλης

Lectures on Databases: section I "Intro", v. 2022.03

by

Data Science Lab. @ Univ. Piraeus (www.datastories.org)



SQL - Structured Query Language



Non procedural – Declarative – Set oriented

Η SQL αποτελεί μια γλώσσα μη διαδικαστική (non-procedural) γλώσσα, καθώς δεν ορίζει το πώς θα βρεθούν οι απαντήσεις. Τα ερωτήματα που δημιουργεί δηλώνουν ποια δεδομένα (ή ποιες αλλαγές) χρειάζεται να επιστραφούν ως αποτέλεσμα (declarative). Μπορεί, έτσι να επιστρέψει πολλές γραμμές (records) δεδομένων αρκεί αυτά να ικανοποιούν τα απαιτούμενα κριτήρια (set oriented).

Αντίθετα στη γλώσσα η οποία είναι διαδικαστική (procedural), τα ερωτήματα (queries) που δημιουργεί ορίζουν ποια και πώς θα αναζητηθούν τα δεδομένα που χρειάζεται να επιστραφούν ως αποτέλεσμα. Μπορεί, έτσι να επιστρέψει μία γραμμή (records/αντικειμένων) δεδομένων. Αυτού του είδους οι γλώσσες μπορούν να λειτουργήσουν ενσωματωμένες σε γλώσσες προγραμματισμού ανωτέρου επιπέδου. (loop, retrieve, process)

SQL



SQL Commands

DDL

- CREATE
- ALTER
- DROP
- TRUNCATE
- RENAME

DML

- SELECT
- INSERT
- UPDATE
- DELETE
- MERGE
- LOCK TABLE

DCL

- GRANT
- REVOKE

TCL

- COMMIT
- ROLLBACK
- SAVEPOINT

ΠΑ.ΠΕΙ. – Καθ. Γιάννης Θεοδωρίδης, Συνδιδάσκουσα: Δρ. Μαυροπόδη Ρόζα 41

SQL ή Sequel

- Αρχικά, SEQUEL (Structured English QUEry Language) για το IBM SYSTEM R. από τους
 Donald D. Chamberlin και Raymond F. Boyce
- Αποτελεί πρότυπο για όλα τα εμπορικά RDBMS.
- Το πρότυπο υλοποιήθηκε από American National Standards Institute (ANSI US) (1986)
 (SQL-86 ή SQL1) και International Standards Organization (ISO EU) (1987)
- SQL-92 ή SQL2
- SQL:1999 ή SQL3, προσθήκη των triggers, object-orientation
- SQL:2003 και SQL:2006, με πολλές προθήκες χαρακτηριστικών XML
- SQL: 2008, χρήση του ORDER BY έξω από κέρσορες, triggers επέκταση, TRUNCATE...
- SQL:2011, βελτιώσεις στη χρήση των προσωρινών (temporal) databases
- SQL:2016 (τελευταία σταθερή έκδοση), περισσότερη υποστήριξη του προτύπου json, row pattern matching ...

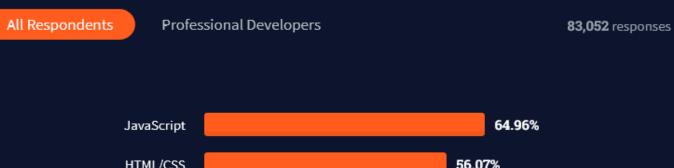
http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt

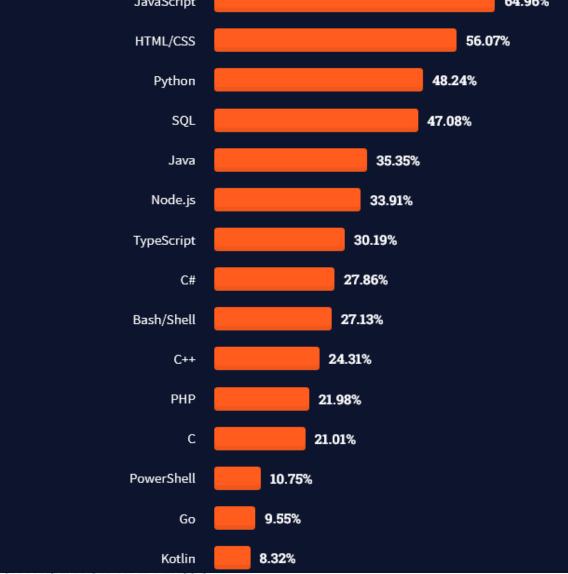
Δημοτικότητα

www.stackoverflow.com 2019

www.stackoverflow.com 2020

www.stackoverflow.com 2021





Γιατί SQL;



www.quora.com

SQL isn't a programming language, and it is often used as part of programs in a range of languages. So we have a cross language tool that makes database access more consistent. Change languages and your SQL stays the same. Change databases and the SQL doesn't change normally much.

stackoverflow.com

I think that the underlying theme today is that rather than replace SQL with one new query language, we are instead creating language-specific frontends to hide the SQL in our regular every-day programming languages and treating SQL as the *protocol* for talking to relational databases.

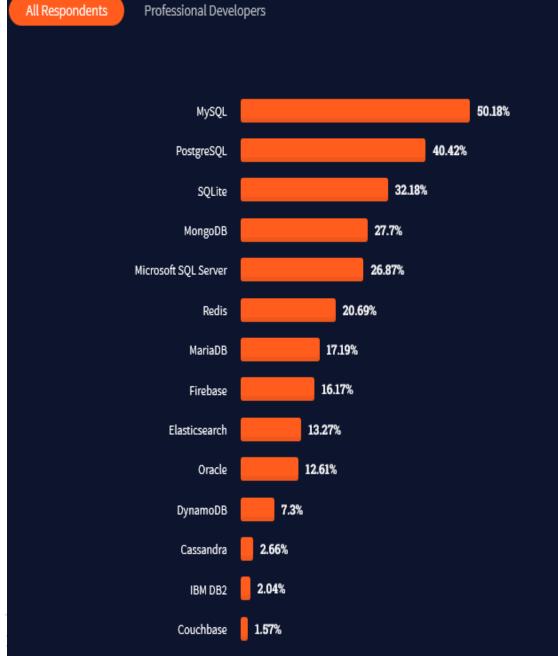
Συνέβαλλε κατά πολύ και η μεγάλη δημοφιλία των RDBMS

Ποια Βάση / DBMS

www.stackoverflow.com 2019

www.stackoverflow.com 2020

www.stackoverflow.com 2021



73,317 responses

SQL-DDL



Η SQL ως γλώσσα DDL περιλαμβάνει τα ρήματα/ λέξεις κλειδιά

- ightharpoonup create: π.χ. create schema/database, create table, create view, create index, create domains ή/και create roll
- \triangleright alter: $\pi.\chi$. alter table, alter database, alter index, alter view
- drop: π.χ. drop table, drop database, drop index, drop view ή/και drop roll



αρχικά δημιουργείται ο ιδιοκτήτης της βάσης

στη συνέχεια δημιουργείται η βάση

και μετά οι σχέσεις (tables)

SQL-DDL create role / user



Core SQL

CREATE ROLE name [WITH ADMIN role_name]

Δημιουργία ρόλου στο σύστημα με τα κατάλληλα δικαιώματα.

Εφαρμογή

CREATE ROLE myrole;

Postgres

```
CREATE ROLE name [ [ WITH ] option [ ... ] ]
where option can be:
      SUPERUSER | NOSUPERUSER
     CREATEDB | NOCREATEDB
     CREATEROLE | NOCREATEROLE
     INHERIT | NOINHERIT
     LOGIN | NOLOGIN
     REPLICATION | NOREPLICATION
     BYPASSRLS | NOBYPASSRLS
      CONNECTION LIMIT connlimit
     [ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL
     VALID UNTIL 'timestamp'
     IN ROLE role_name [, ...]
      ROLE role_name [, ...]
     ADMIN role name [, ...]
     SYSID uid
```

ΒΔ: [3] Η Γλώσσα SQL

– Καθ. Γιάννης Θεοδωρίδης, Συνδιδάσκουσα: Δρ. Μαυρ<mark>οπόδη Ρόζα</mark>

SQL-DDL create role 1-2



CREATE ROLE datamaster **CREATEDB CREATEROLE**;

CREATE ROLE datamaster **LOGIN PASSWORD** 'mypass123';

postgres $(\pi.\chi)$

CREATE ROLE datamaster SUPERUSER LOGIN PASSWORD 'mypass123' VALID UNTIL '2022-09-31';

SQL σε Postgres RDBMS



psql (command line)

PgAdmin GUI

SQL-DDL create database 1-1



Core SQL

CREATE DATABASE database_name;

Δημιουργία της βάσης.

Το πρότυπο ANSI δεν περιέχει την create database, αλλά το create schema.

[database_name]: το πλήθος των χαρακτήρων εξαρτάται από το ΔΒΜΣ Ανάλογα το ΛΣ τα ονόματα μπορεί να είναι case-sensitive ή όχι

Εφαρμογή

create database mydatabase;

Postgres

SQL-DDL create database 1-2



CREATE DATABASE myunipi;

CREATE DATABASE myunipi **OWNER** datamaster;

Postgres $(\pi.\chi.)$

CREATE DATABASE myunipi **WITH OWNER**=datamaster **ENCODING**=utf8;

CREATE DATABASE greek ENCODING 'utf-8'
LC_COLLATE='el_GR.utf8' LC_CTYPE='el_GR.utf8'
WITH OWNER=datamaster;
/*πρέπει να υπάρχουν τα αντίστοιχα locales*/



Core SQL

```
CREATE TABLE table_name(
column1 data_type[(size)][column_constraints],
column2 data_type[(size)],
[table_constraints]....);
```

Εφαρμογή

```
create table department
  (dept_name varchar(20),
  building varchar(15),
  budget numeric(12,2) check (budget > 0),
primary key (dept_name) );
```



Postgres (τμήμα του ορισμού)

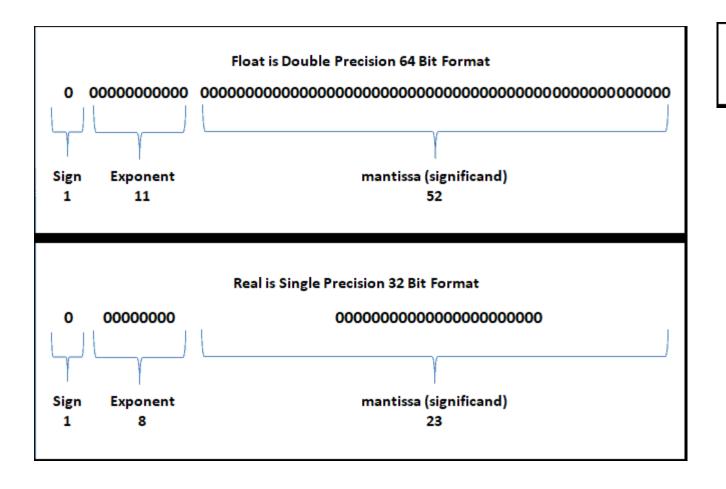
```
CREATE [ { TEMPORARY | TEMP } ] TABLE [ IF NOT EXISTS ]
   table_name
  { column_name data_type [ COLLATE collation ] [ column_constraint [ ... ] ]
     table_constraint
     LIKE ..... }
   [, ... ]
[ INHERITS ( parent_table [, ... ] ) ]
[ PARTITION ......]
[ USING ......]
[ WITH .....]
[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]
[ TABLESPACE tablespace_name ]
```

SQL DDL τύποι δεδομένων

- Αριθμητικά (numeric): ακέραιους αριθμούς διαφόρων μεγεθών (INTEGER ή INT και SMALLINT), αριθμούς κινητής υποδιαστολής διαφορετικής ακρίβειας (REAL ή FLOAT SINGLE ή DOUBLE PRECISION). Οι μορφοποιημένοι αριθμοί μπορούν να δηλωθούν χρησιμοποιώντας DECIMAL(i, j)—ή DEC(i, j) ή NUMERIC(i, j).
- Χαρακτήρων (character-string): CHAR(n) ή CHARACTER(n), όπου n είναι ο αριθμός των χαρακτήρων, μεταβαλλόμενο μήκος— VARCHAR(n) ή CHAR VARYING(n) ή CHARACTER VARYING(n), όπου n είναι ο μέγιστος αριθμός χαρακτήρων, συμβολοσειράς μεταβλητού μήκους που ονομάζεται CHARACTER LARGE OBJECT ή CLOB είναι επίσης διαθέσιμος για τον καθορισμό στηλών που έχουν μεγάλες τιμές κειμένου, όπως έγγραφα. Το μέγιστο μήκος CLOB μπορεί να καθοριστεί σε kilobyte (K), megabyte (M) ή gigabyte (G). Για παράδειγμα, το CLOB(20M) καθορίζει ένα μέγιστο μήκος 20 megabyte.

SQL DDL τύποι δεδομένων





Significand $x\ base^{\text{exponent}}$

SQL DDL τύποι δεδομένων

- Λογικά δεδομένα: Ένας τύπος δεδομένων boolean έχει τις παραδοσιακές τιμές TRUE ή FALSE. Στην SQL, λόγω της παρουσίας τιμών NULL, χρησιμοποιείται μια λογική τριών τιμών, επομένως μια τρίτη πιθανή τιμή για έναν τύπο δεδομένων Boolean είναι UNKNOWN.
- Ημερομηνίας ώρας: Ο τύπος δεδομένων DATE έχει δέκα θέσεις και τα στοιχεία του είναι YEAR, MONTH και DAY με τη μορφή EEEE-MM-HH. Ο τύπος δεδομένων TIME έχει τουλάχιστον οκτώ θέσεις, με τα στοιχεία HOUR, MINUTE και SECOND στη μορφή ΩΩ:ΛΛ:ΔΔ. Timestamp: Ένας τύπος δεδομένων χρονικής σήμανσης (TIMESTAMP) περιλαμβάνει τα πεδία DATE και TIME. Interval (relative value)
- και πολλά περισσότερα όπως, serial, geometric, network IP (ipv4, ipv6, mac)
 κλπ



PostgreSQL ∼ Data Types

Name	Example
Αριθμητικά δεδομένα (Numeric types)	1 , 1.45353553, 342522524324.5
Δεδομένα χαρακτήρων (character types)	'ok', 'hello world'
Λογικά δεδομένα (Boolean types)	FALSE, TRUE
Δεδομένα ημερομηνίας/ώρας Date/Time types	1999-01-01
Γεωμετρικά δεδομένα (Geometric types)	18.2700 33.2300

ΠΑ.ΠΕΙ. – Καθ. Γιάννης Θεοδωρίδης, Συνδιδάσκουσα: Δρ. Μαυροπόδη Ρόζα



Η ημερομηνία 20220320 πώς πραγματικά αποθηκεύεται στη βάση; Όχι πώς φαίνεται π.χ. είναι char; varchar; Μήπως είναι int;



fname char(5) /* περιέχει το όνομα rosa */
Iname varchar(5) /*περιέχει το όνομα rosa*/

Η ερώτηση <fname είναι ίσο με Iname> τί αποτέλεσμα θα έχει ??

SQL-DDL user datatype create domain



CREATE DOMAIN my_domain **AS VARCHAR(10)**;

Postgres $(\pi.\chi.)$

CREATE DOMAIN my_domain AS VARCHAR(10) CHECK (VALUE IN ('white', 'red', 'rose', 'sparkling'));

CREATE DOMAIN my_domain **AS VARCHAR(1) CHECK** (**VALUE** ~ '^[a-f]\$'); /*δέχεται μόνο πεζούς χαρακτήρες a-f case sensitive*/

SQL-DDL constrains



Η SQL υποστηρίζει μια πληθώρα από περιορισμούς, όπως:

- **NOT NULL**: ενώ το NULL είναι τιμή το NOT NULL αποτελεί περιορισμό χαρακτηριστικού (attribute constraint)
- UNIQUE: μοναδικό
- **DEFAULT**: απόδοση αρχικών προκαθορισμένων τιμών π.χ. grade INT DEFAULT 10,
- CHECK: έλεγχος
 salary NUMERIC(8,2) CHECK (salary > 29000) ή
 col1 VARCHAR(10) CHECK (VALUE IN ('white', 'red', 'rose', 'sparkling'));
 col2 CHAR(5) CHECK (col2 ~ '^[α-ω]{5}\$');/*regular expressions*/
 PRIMARY KEY: πρωτεύον κλειδί (not null και unique, 1+ columns)
- FOREIGN KEY: ξένο κλειδί. (on delete/update SET NULL/CASCADE/SET DEFAULT deferred)

SQL-DDL create table constraints



```
create table student
                                         varchar(5),
                   (ID
                                         varchar(20) not null,
                   name
                   dept_name
                                         varchar(20),
(\pi.\chi.)
                                         numeric(3,0) check (tot cred >= 0),
                   tot cred
                   primary key (ID),
                   foreign key (dept_name) references department (dept_name)
                          on delete set null
```

Ερώτηση: Τί τύπος δεδομένων είναι το dept_name στο table department;

SQL-DDL create table constraints names



```
create table student
                     varchar(5),
       (ID
                     varchar(20) CONSTRAINT name_not_null not null,
       name
       dept_name varchar(20),
(\pi.\chi.)
                     numeric(3,0) CONSTRAINT totcred_over0 check (tot_cred >= 0),
       tot cred
       primary key (ID),
       CONSTRAINT fk_deptname foreign key (dept_name) references department
       (dept name)
              on delete set null
```



Ένα πρωτεύον κλειδί τι τύπος δεδομένων (π.χ. char, varchar, int κλπ) είναι σε μια βάση;

SQL-DDL create table constraint-default



 $(\pi.\chi.)$

```
CREATE TABLE mytable

ID varchar(5),

course_id varchar(8),

sec_id archar(8),

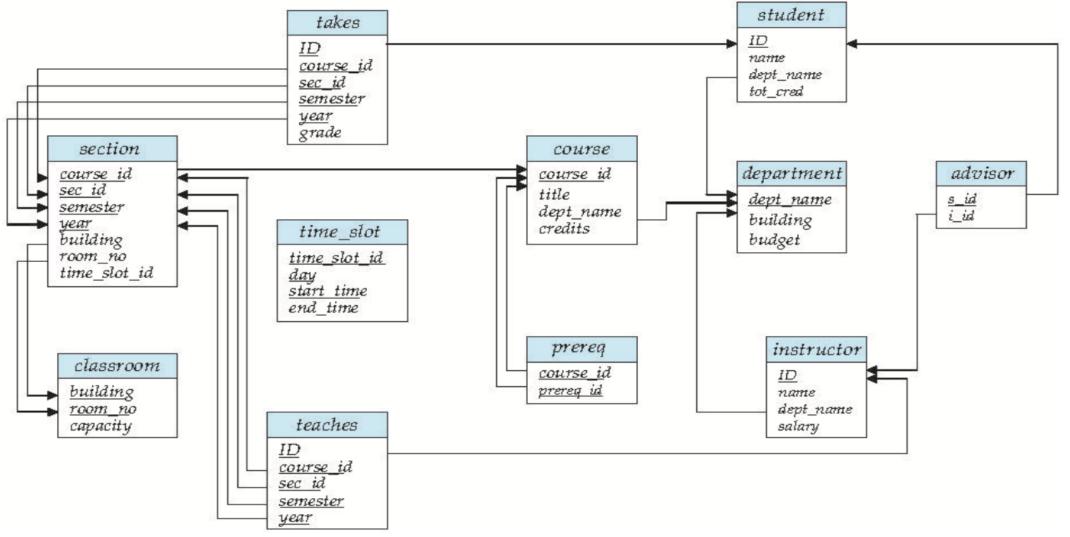
semester varchar(6),

year numeric(4,0),

grade int DEFAULT 10

);
```







primary key

insert into department values ('Biology', 'Watson', '90000'); insert into department values ('Comp. Sci.', 'Taylor', '100000'); insert into department values ('Elec. Eng.', 'Taylor', '85000');

(π.χ.)

psql>/d department
select * from department;

(unique)
insert into department
values ('Biology', 'Central', '1');

(not null)
insert into department
values ('Biology', 'Central', '1');

```
mypostgres=# insert into department values ('Biology', 'Central', '1');
ERROR: duplicate key value violates unique constraint "department_pkey"
DETAIL: Key (dept_name)=(Biology) already exists.
mypostgres=# _______
```

```
mypostgres=# insert into department values (null, 'Central', '1');
ERROR: null value in column "dept_name" of relation "department" violates not-null constraint
DETAIL: Failing row contains (null, Central, 1.00).
mypostgres=#
```



foreign key

insert into student values ('00128', 'Zhang', 'Biology', '102'); insert into student values ('12345', 'Shankar', 'Comp. Sci.', '32'); insert into student values ('19991', 'Brandt', 'Biology', '80');

 $(\pi.\chi.)$

select * from student;

delete from student where id='00128'; /*κανένα πρόβλημα*/



foreign key

(2 rows)

```
delete from department where dept name='Biology';
/* Γιατί έχει κενό (null) στη δεύτερη γραμμή του πίνακα student; */
 mypostgres=# delete from department where dept name='Biology';
 DELETE 1
 mypostgres=# select * from department;
 Active code page: 1253
                              budget
   dept name | building |
   Comp. Sci. | Taylor
                           100000.00
   Elec. Eng. | Taylor
                              85000.00
  (2 rows)
 mypostgres=# select * from student;
 Active code page: 1253
     id
                      dept name
                                 tot cred
             name
            Shankar
                      Comp. Sci.
    12345
    19991
            Brandt
```

ΒΔ: [3] Η Γλώσσα SQL

 $(\pi.\chi.)$

SQL-DDL create table generated as identity



postgres

```
create table department
  (dept_name varchar(20),
  building varchar(15),
  budget numeric(12,2) check (budget > 0),
  dept_id int generated { always | by default } as identity
  primary key (dept_name) );
```

SQL-DDL create table generated always as ..



postgres

```
create table department
  (dept_name varchar(20),
  building varchar(15),
  budget numeric(12,2) check (budget > 0),
  double_budget int generated always as (budget * 2) stored
  primary key (dept_name) );
```

SQL-DDL drop role 1-1



Core SQL

DROP ROLE name ;

Εφαρμογή

DROP ROLE datamaster;

Postgres

DROP ROLE [**IF EXISTS**] name [, ...]

DROP ROLE IF EXISTS myrole;

DROP ROLE myrole1, myrole2;

SQL-DDL drop table 1-1



Core SQL

DROP TABLE name [CASCADE | RESTRICT];

Εφαρμογή

DROP TABLE student;

Postgres

DROP TABLE [**IF EXISTS**] name [, ...] [**CASCADE** | **RESTRICT**]

SQL-DDL drop table 1-2



DROP TABLE students;

DROP TABLE IF EXISTS students;

DROP TABLE students, department;

DROP TABLE department **CASCADE**;

DROP TABLE department **RESTRICT**;

 $(\pi.\chi.)$

SQL-DDL drop database 1-1



Core SQL

DROP DATABASE name;

Δεν υπάρχει στο πρότυπο.

Εφαρμογή

DROP DATABASE mydatabase;

Postgres

DROP DATABASE [IF EXISTS] name [[WITH] (FORCE)]

DROP DATABASE IF EXISTS mydatabase;

DROP DATABASE mydatabase1;

SQL-DDL alter table 1-1



Core SQL

```
ALTER TABLE table name
[action ...];
όπου action μπορεί να είναι π.χ:
ADD [ COLUMN ]
DROP [ COLUMN ]
ALTER [ COLUMN ] [ SET | DROP ] DEFAULT
ALTER [ COLUMN ] [ SET | DROP ] NOT NULL
ADD CONSTRAINT
RENAME RENAME [COLUMN] existing col name
```

TO new col name

SQL-DDL alter table 1-2

ALTER TABLE student ADD COLUMN col_name VARCHAR(25);
ALTER TABLE student DROP COLUMN col_name;

ALTER TABLE student

ALTER COLUMN existing_col_name TYPE VARCHAR(20), ADD COLUMN col_name2 VARCHAR(10);

ALTER TABLE student RENAME COLUMN existing_col_name TO

π.χ. new_col_name;

ALTER TABLE department ADD COLUMN description TEXT CHECK (description <>'');

ALTER TABLE department RENAME TO mydepartment;

SQL-DDL alter table constraints 1-3



ALTER TABLE department ADD CHECK (name <> ''); ALTER TABLE department ADD CONSTRAINT some_name UNIQUE (product_no);

π.χ.

ALTER TABLE department ADD FOREIGN KEY (dept_group_id) REFERENCES sxoli;

ALTER TABLE department ADD CONSTRAINT some_name FOREIGN KEY (dept_group_id) REFERENCES sxoli;

ALTER TABLE department DROP CONSTRAINT some_name;

SQL-DDL alter table constraints 1-4

ALTER TABLE department ALTER COLUMN dept_no SET NOT NULL; ALTER TABLE department ALTER COLUMN existing_col_name DROP NOT NULL;

π.χ.

ALTER TABLE department ALTER COLUMN space SET DEFAULT 50; ALTER TABLE department ALTER COLUMN space DROP DEFAULT 50;

ALTER TABLE department ALTER COLUMN space TYPE NUMERIC(10,2); ALTER TABLE department ALTER COLUMN space TYPE VARCHAR(10);

Postgres --psql connect to



```
Command Prompt - psql -U postgres
```

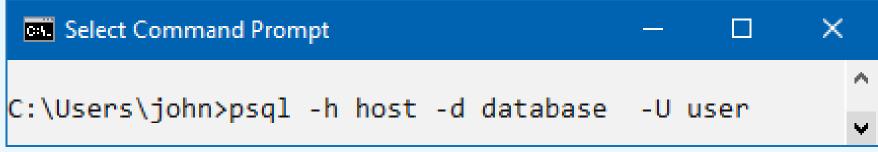
Microsoft Windows [Version 10.0.17134.1902] (c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\john>psql -U postgres
Password for user postgres: _

```
postgres=# \?
General
\copyright show PostgreSQL
\crosstabview [COLUMNS] execute query
\errverbose show most recer
```

Postgres --psql connect to remote host database





psql

Η βάση βρίσκεται σε απομακρυσμένο υπολογιστή

Postgres --psql list all databases



·	age: 1253		List of databa	ses		
Name Owner Er		Encoding Collate		Ctype	+ Access privileges	
nypostgres personal	postgres postgres postgres	UTF8 UTF8	English_United States.1253 English_United States.1253	English_United States.1253 English_United States.1253	 =Tc/postgres + postgres=CTc/postgres+ rosa=CTc/postgres	
oostgres template0	postgres postgres 	UTF8 UTF8	English_United States.1253 English_United States.1253	English_United States.1253 English_United States.1253	 =c/postgres + postgres=CTc/postgres	
emplate1	postgres	UTF8	English_United States.1253	English_United States.1253	=c/postgres + postgres=CTc/postgres	

Postgres --psql connect to other database



```
mydatabase-#
mydatabase-#
mydatabase-#
mydatabase-# \c mydadtabse2
FATAL: database "mydadtabse2" does not exist
Previous connection kept
mydatabase-# \c mydatabase2
You are now connected to database "mydatabase2" as user "postgres".
mydatabase2-# _
```

Postgres --psql list all tables/relations



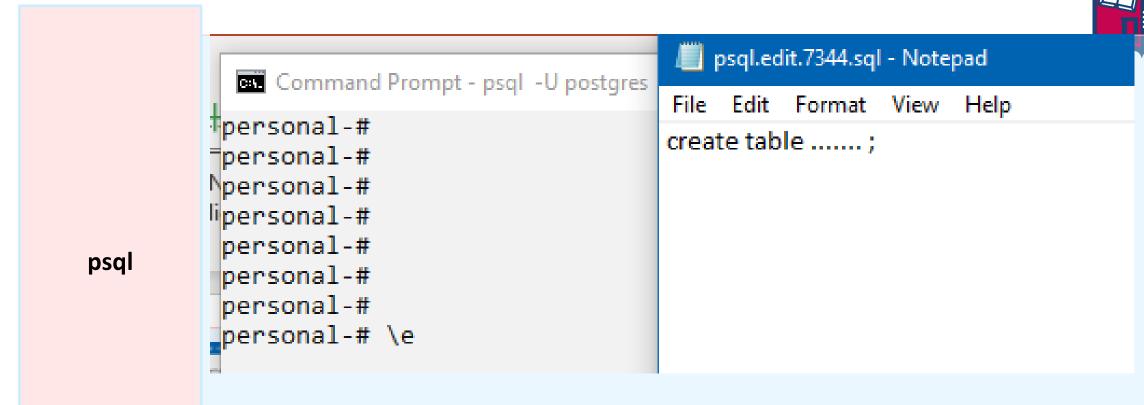
	List of re				
Schema	Name	Туре	Owner		
public	classroom	table	postgres		
public	course	table			
public	•	•	_		
public	instructor				
public	me .	table			
•	section	table	postgres		
6 rows)					

Postgres --psql details of a table



```
Command Prompt - psql -U postgres
                                                                           personal-# \d department
                    Table "public.department"
                   Type | Collation | Nullable | Default
 Column
 dept name | character varying(20) |
                                             | not null
 building | character varying(15)
 budget | numeric(12,2)
Indexes:
    "department_pkey" PRIMARY KEY, btree (dept_name)
Check constraints:
    "department budget check" CHECK (budget > 0::numeric)
Referenced by:
   TABLE "instructor" CONSTRAINT "instructor_dept_name_fkey" FOREIGN KEY (dept_name)
 REFERENCES department(dept name) ON DE-- More -- _
```

Postgres --psql work with editor



Postgres --psql useful



psql

\l λίστα με όλες τις βάσεις που υπάρχουν στο ΣΔΒΔ (list) \c database σύνδεση με τη συγκεκριμένη βάση (connect) \dt προβολή των σχέσεων της βάσης (database tables) \d table περιγραφή του ορισμού της σχέσης (description) \e ανοίγει editor για την επεξεργασία των ερωτημάτων (editor) \h some-SQL-command πληροφορίες για τις εντολές SQL (help) \dn List available schema (database schema-nodes) \du List users and their roles (database users) \g Execute the previous command (go)

\timing Turn on/off query execution time

\q quit