

Βάσεις Δεδομένων Databases



2^η Εργαστηριακή Διάλεξη: PostgreSQL

DataStories (datastories.org)

Τμήμα Πληροφορικής
Πανεπιστήμιο Πειραιώς
(4^ο εξ.)

Βοηθοί Εργαστηρίου

Πέτρος Πέτρου
ppetrou@unipi.gr

Γιάννης Κοντούλης
ikontoulis@unipi.gr

Outline



DataStories

- *Hands on (DB)
- Composite Types, Natural Join
- Rename, Like, Order by
- Union, Intersect, Except
- Aggregate Functions, Nested Queries
- Joins Expressions
- *Hands on (Queries)



Outline



DataStories

- ***Hands on (DB)**
- Composite Types, Natural Join
- Rename, Like, Order by
- Union, Intersect, Except
- Aggregate Functions, Nested Queries
- Joins Expressions
- *Hands on (Queries)



Connect to server



DataStories

pgAdmin 4

File Object Tools Help

Dashboard Properties SQL Statistics Dependencies Dependents

Welcome

Feature rich | pgAdmin is an Open Source tool for managing PostgreSQL databases.

Quick Links

Getting Started

PostgreSQL Database Manager

Create - Server

General Connection SSL Advanced

Host name/address

Port

Maintenance database

Username

Password

Save password?

Role

Name must be specified.

Save Cancel Reset

195.251.230.8

45432

postgres

postgres

students

infol@bdbl@bs2017

Configure pgAdmin

Get PostgreSQL

Community Support



*Hands on Table City [1/2]



Attributes:

- ▶ **ID:** Σειριακός (αύξων) ακέραιος (not null).
- ▶ **Name:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **CountryCode:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **District:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **Population:** Ακέραιος, προκαθορισμένη τιμή 0 (not null).
- ▶ **PK: ID.**

*Hands on Table City [2/2]



DataStories

```
DROP TABLE IF EXISTS City;  
CREATE TABLE City (  
  ID serial NOT NULL,  
  Name text NOT NULL,  
  CountryCode text NOT NULL,  
  District text NOT NULL,  
  Population integer NOT NULL default 0,  
  PRIMARY KEY (ID)  
);
```

*Hands on Table Country [1/4]



Attributes:

- ▶ **Code:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **Name:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **Continent:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **Region:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **SurfaceArea:** Διπλής ακρίβειας, προκαθορισμένη τιμή 0,00 (not null).
- ▶ **IndepYear:** Ακέραιος, προκαθορισμένη τιμή null.

*Hands on Table Country [2/4]



Attributes:

- **Population:** Ακέραιος, προκαθορισμένη τιμή 0 (not null).
- **LifeExpectancy:** Πραγματικός, προκαθορισμένη τιμή null.
- **GNP:** Διπλής ακρίβειας, προκαθορισμένη τιμή null.
- **GNPOld:** Διπλής ακρίβειας, προκαθορισμένη τιμή null.
- **LocalName:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).

*Hands on

Table Country [3/4]



Attributes:

- ▶ **GovernmentForm:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **HeadOfState:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους, προκαθορισμένη τιμή null.
- ▶ **Capital:** Ακέραιος, προκαθορισμένη τιμή null.
- ▶ **Code2:** Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **PK: Code.**

*Hands on

Table Country [4/4]



DataStories

DROP TABLE IF EXISTS Country;

CREATE TABLE Country (

Code **text NOT NULL**,

Name **text NOT NULL**,

Continent **text NOT NULL**,

Region **text NOT NULL**,

SurfaceArea **double precision NOT NULL default 0.00**,

IndepYear **integer default NULL**,

Population **integer NOT NULL default 0**,

LifeExpectancy **real default NULL**,

GNP **double precision default NULL**,

GNPOld **double precision default NULL**,

LocalName **text NOT NULL**,

GovernmentForm **text NOT NULL**,

HeadOfState **text default NULL**,

Capital **integer default NULL**,

Code2 **text NOT NULL**,

PRIMARY KEY (Code)

);

*Hands on

Table CountryLanguage [1/2]



Attributes:

- ▶ **CountryCode**: Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **Language**: Συμβολοσειρά χαρακτήρων μεταβλητού απεριόριστου μήκους (not null).
- ▶ **IsOfficial**: **official type**, προκαθορισμένη τιμή 'F' (not null).
- ▶ **Percentage**: Πραγματικός, προκαθορισμένη τιμή 0,0 (not null).
- ▶ PRIMARY KEY (**CountryCode**, **Language**).

*Hands on

Table CountryLanguage [2/2]



DataStories

```
CREATE TYPE official AS ENUM ('T','F');
```

```
DROP TABLE IF EXISTS CountryLanguage;
```

```
CREATE TABLE CountryLanguage (
```

```
CountryCode text NOT NULL,
```

```
Language text NOT NULL,
```

```
IsOfficial official NOT NULL default 'F',
```

```
Percentage real NOT NULL default 0.0,
```

```
PRIMARY KEY (CountryCode,Language)
```

```
);
```

Outline



DataStories

- *Hands on (DB)
- **Composite Types, Natural Join**
- Rename, Like, Order by
- Union, Intersect, Except
- Aggregate Functions, Nested Queries
- Joins Expressions
- *Hands on (Queries)



Εισαγωγή Δεδομένων [1/2]

Insert



DataStories

```
INSERT INTO table_name VALUES (...);
```

✓ Παράδειγμα:

```
INSERT INTO City VALUES (4042, 'Miami Beach', 'USA', 'Florida', 97855);
```

```
INSERT INTO City (ID, Name, CountryCode, District, Population)  
VALUES (4042, 'Miami Beach', 'USA', 'Florida', 97855);
```

Εισαγωγή Δεδομένων [2/2]

Copy



DataStories

```
COPY table_name FROM path;
```

✓ Παράδειγμα:

```
COPY City FROM '/private/tmp/country_data.csv';
```

```
COPY City FROM '/private/tmp/country_data.csv' delimiters ',' csv header;
```

Σύνθετοι Τύποι Δεδομένων (Composite types) [1/2]

Create Type



- Τύποι δεδομένων που ορίζονται από τον χρήστη (Object-Relational).
- Only **field names & types** can be specified.
- **No constraints** (e.g. NOT NULL) can presently be included.
- The **AS** keyword is essential.

```
CREATE TYPE inventory_item AS (  
  name text,  
  supplier_id integer,  
  price numeric  
);
```

```
CREATE TABLE on_hand (  
  item inventory_item,  
  price numeric  
);
```


Insert - Access

- Τύποι δεδομένων που ορίζονται από τον χρήστη (Object-Relational).

```
INSERT INTO on_hand VALUES (ROW ('fuzzy dice', 42, 1.99), 1000);
```

```
SELECT (item).name  
FROM on_hand  
WHERE (item).price > 9.99;
```

```
SELECT (on_hand.item).name  
FROM on_hand  
WHERE (on_hand.item).price > 9.99;
```

Βασική Δομή SQL Ερωτημάτων [1/3]

Φυσική Σύνδεση (natural join)



- Λειτουργεί με δύο σχέσεις & παράγει μια σχέση σαν αποτέλεσμα.
- Εξετάζει μόνο αυτά τα ζευγάρια εγγραφών, που έχουν την ίδια τιμή σε εκείνα τα χαρακτηριστικά που εμφανίζονται στα σχήματα & των δύο σχέσεων.
- **Σειρά Εμφάνισης:**
 - ▶ Χαρακτηριστικά που είναι κοινά για τα σχήματα & των δύο σχέσεων (τα παραθέτουμε μόνο μια φορά).
 - ▶ Χαρακτηριστικά που είναι μοναδικά στο σχήμα της πρώτης σχέσης.
 - ▶ Χαρακτηριστικά που είναι μοναδικά στο σχήμα της δεύτερης σχέσης.

Βασική Δομή SQL Ερωτημάτων [2/3]

Φυσική Σύνδεση (natural join)



- ✓ Για όλους τους καθηγητές στο Πανεπιστήμιο που έχουν διδάξει κάποιο μάθημα, βρείτε τα ονόματά τους & το ID του μαθήματος όλων των μαθημάτων που δίδαξαν.

```
SELECT name, course_id  
FROM instructor, teaches  
WHERE instructor.ID = teaches.ID;
```

```
SELECT name, course_id  
FROM instructor NATURAL JOIN teaches;
```

Βασική Δομή SQL Ερωτημάτων [3/3]

Φυσική Σύνδεση (natural join)



- ✓ Αναφέρατε τα ονόματα των καθηγητών μαζί με τους τίτλους των μαθημάτων που διδάσκουν.

```
SELECT name, title  
FROM instructor NATURAL JOIN teaches, course  
WHERE teaches.course_id = course.course_id;
```

```
SELECT name, title  
FROM instructor NATURAL JOIN teaches NATURAL JOIN course;
```

```
SELECT name, title  
FROM (instructor NATURAL JOIN teaches) JOIN course USING (course_id);
```

Outline



DataStories

- *Hands on (DB)
- Composite Types, Natural Join
- **Rename, Like, Order by**
- Union, Intersect, Except
- Aggregate Functions, Nested Queries
- Joins Expressions
- *Hands on (Queries)



Πρόσθετες Απλές Πράξεις

Η Πράξη Μετονομασίας (Rename) [1/2]



✓ **SELECT, FROM**

- ▶ Δύο σχέσεις στον όρο `from` μπορεί να έχουν χαρακτηριστικά με το ίδιο όνομα, οπότε ένα όνομα χαρακτηριστικού αντιγράφεται στο αποτέλεσμα.
- ▶ Αν χρησιμοποιήσουμε μια αριθμητική έκφραση στη `select`, το τελικό χαρακτηριστικό δεν θα έχει όνομα.
- ▶ Για να αλλάξουμε το όνομα του χαρακτηριστικού στο αποτέλεσμα.
- ▶ Για να αντικατασταθεί ένα μακροσκελές όνομα σχέσης με μία μικρότερη εκδοχή του.
- ▶ Στην περίπτωση που θέλουμε να συγκρίνουμε εγγραφές απ' την ίδια σχέση.

Πρόσθετες Απλές Πράξεις

Η Πράξη Μετονομασίας (Rename) [2/2]

```
SELECT name AS instructor_name, course_id  
FROM instructor, teaches  
WHERE instructor.ID = teaches.ID;
```

```
SELECT T.name, S.course_id  
FROM instructor AS T, teaches AS S  
WHERE T.ID = S.ID;
```

```
SELECT DISTINCT T.name  
FROM instructor AS T, instructor AS S  
WHERE T.salary > S.salary AND S.dept_name = 'Biology';
```

Πρόσθετες Απλές Πράξεις

Πράξεις με Συμβολοσειρές [1/3]



✓ WHERE

- ▶ Ο χαρακτήρας '%' ταιριάζει με οποιαδήποτε **υπο-συμβολοσειρά**.
- ▶ Ο χαρακτήρας '_' ταιριάζει με οποιονδήποτε **χαρακτήρα**.
- ▶ Τα μοτίβα είναι ευαίσθητα σε κεφαλαία-πεζά (case sensitive).
- ▶ Η SQL εκφράζει τα μοτίβα χρησιμοποιώντας τον τελεστή σύγκρισης **like**.

Πρόσθετες Απλές Πράξεις

Πράξεις με Συμβολοσειρές [2/3]



✓ Παραδείγματα:

- **'Intro%'**: ταιριάζει με οποιαδήποτε συμβολοσειρά ξεκινά με **'Intro'**.
- **'%Comp%'**: ταιριάζει με οποιαδήποτε συμβολοσειρά περιέχει το **'Comp'** ως υπο-συμβολοσειρά.
- **'___'**: ταιριάζει με οποιαδήποτε συμβολοσειρά έχει ακριβώς τρεις χαρακτήρες.
- **'___%'**: ταιριάζει με οποιαδήποτε συμβολοσειρά έχει τουλάχιστον τρεις χαρακτήρες.

Πρόσθετες Απλές Πράξεις

Πράξεις με Συμβολοσειρές [3/3]



DataStories

✓ Παραδείγματα:

```
SELECT dept_name  
FROM department  
WHERE building LIKE '%Watson%';
```

City (ID, Name, CountryCode, District, Population)

```
SELECT Name, Population  
FROM City  
WHERE Name LIKE 'A%';
```

```
SELECT Name, Population  
FROM City  
WHERE Name LIKE '_c%';
```

```
(SELECT Name, Population  
FROM City  
WHERE Name LIKE 'A%')  
UNION  
(SELECT Name, Population  
FROM City  
WHERE Name LIKE 'B%');
```

Πρόσθετες Απλές Πράξεις

Διάταξη των Πλειάδων

```
SELECT name  
FROM instructor  
WHERE dept_name = 'Physics'  
ORDER BY name;
```

```
SELECT *  
FROM instructor  
ORDER BY salary DESC, name ASC;
```

```
SELECT name  
FROM City  
ORDER BY name;
```

Πρόσθετες Απλές Πράξεις

Κατηγορήματα του όρου WHERE



DataStories

```
SELECT name  
FROM instructor  
WHERE salary >= 90000 AND salary <= 100000;
```

```
SELECT name  
FROM instructor  
WHERE salary BETWEEN 90000 AND 100000;
```

Outline



DataStories

- *Hands on (DB)
- Composite Types, Natural Join
- Rename, Like, Order by
- **Union, Intersect, Except**
- Aggregate Functions, Nested Queries
- Joins Expressions
- *Hands on (Queries)



Πράξεις Συνόλων [1/7]

Ο Τελεστής Ένωσης (union)



- Η πράξη **union** απαλείφει αυτόματα τα διπλότυπα (αντίθετα από τον όρο **select**).
- ✓ Σύνολο όλων των μαθημάτων που διδάχθηκαν είτε το Φθινόπωρο του 2009 είτε την Άνοιξη του 2010, είτε και τα δύο.

```
(SELECT course_id  
FROM section  
WHERE semester = 'Fall' AND year = 2009)  
UNION  
(SELECT course_id  
FROM section  
WHERE semester = 'Spring' AND year = 2010);
```

Πράξεις Συνόλων [2/7]

Ο Τελεστής Ένωσης (union)

- ✓ Σύνολο όλων των μαθημάτων που διδάχθηκαν είτε το Φθινόπωρο του 2009 είτε την Άνοιξη του 2010, είτε και τα δύο (με διπλότυπα).

```
(SELECT course_id
FROM section
WHERE semester = 'Fall' AND year = 2009)
UNION ALL
(SELECT course_id
FROM section
WHERE semester = 'Spring' AND year = 2010);
```

Πράξεις Συνόλων [3/7]

Ο Τελεστής Τομής (intersect)



- Η πράξη **intersect** απαλείφει αυτόματα τα διπλότυπα.
- ✓ Σύνολο όλων των μαθημάτων που διδάχθηκαν το Φθινόπωρο του 2009, καθώς επίσης και την Άνοιξη του 2010.

```
(SELECT course_id  
FROM section  
WHERE semester = 'Fall' AND year = 2009)  
INTERSECT  
(SELECT course_id  
FROM section  
WHERE semester = 'Spring' AND year = 2010);
```


Πράξεις Συνόλων [4/7]

Ο Τελεστής Τομής (intersect)

- ✓ Σύνολο όλων των μαθημάτων που διδάχθηκαν το Φθινόπωρο του 2009, καθώς επίσης και την Άνοιξη του 2010 (με διπλότυπα).

```
(SELECT course_id  
FROM section  
WHERE semester = 'Fall' AND year = 2009)  
INTERSECT ALL  
(SELECT course_id  
FROM section  
WHERE semester = 'Spring' AND year = 2010);
```

Πράξεις Συνόλων [5/7]

Ο Τελεστής Διαφοράς (except)



- Η πράξη **except** εξάγει όλες τις εγγραφές από την πρώτη σχέση που δεν υπάρχουν στη δεύτερη σχέση (εκτελεί διαφορά συνόλων).
- Απαλείφει αυτόματα τα διπλότυπα στις σχέσεις πριν εκτελέσει τη διαφορά.

Πράξεις Συνόλων [6/7]

Ο Τελεστής Διαφοράς (except)

- ✓ Όλα τα μαθήματα που διδάχθηκαν το Φθινόπωρο του 2009, αλλά όχι την Άνοιξη του 2010.

```
(SELECT course_id  
FROM section  
WHERE semester = 'Fall' AND year = 2009)  
EXCEPT  
(SELECT course_id  
FROM section  
WHERE semester = 'Spring' AND year = 2010);
```

Πράξεις Συνόλων [7/7]

Ο Τελεστής Διαφοράς (except)

- ✓ Όλα τα μαθήματα που διδάχθηκαν το Φθινόπωρο του 2009, αλλά όχι την Άνοιξη του 2010 (με διπλότυπα).

```
(SELECT course_id  
FROM section  
WHERE semester = 'Fall' AND year = 2009)  
EXCEPT ALL  
(SELECT course_id  
FROM section  
WHERE semester = 'Spring' AND year = 2010);
```

Outline



DataStories

- *Hands on (DB)
- Composite Types, Natural Join
- Rename, Like, Order by
- Union, Intersect, Except
- **Aggregate Functions, Nested Queries**
- Joins Expressions
- *Hands on (Queries)



Συναθροιστικές Συναρτήσεις (Aggregate Functions)

SELECT, HAVING



✓ SELECT, HAVING

- Συναρτήσεις που παίρνουν ως είσοδο μία συλλογή τιμών (σύνολο/πολλαπλό σύνολο) & επιστρέφουν μία μόνο συνάρτηση.
- Οι συναθροιστικές συναρτήσεις αγνοούν τις **null** τιμές.
- Ενσωματωμένες Συναθροιστικές Συναρτήσεις :

avg, min, max, sum, count

Συναθροιστικές Συναρτήσεις (Aggregate Functions)

Απλές Συνοπτικές Συναρτήσεις



```
SELECT COUNT (*)  
FROM course;
```

```
SELECT AVG (salary) AS avg_salary  
FROM instructor  
WHERE dept_name = 'Biology';
```

```
SELECT COUNT (DISTINCT ID)  
FROM teaches  
WHERE semester = 'Spring' AND year = 2010;
```

Συναθροιστικές Συναρτήσεις (Aggregate Functions)

Ομαδοποίηση Εγγραφών (group by) [1/2]



- Το χαρακτηριστικό/χαρακτηριστικά που δίνονται στον όρο **group by** χρησιμοποιούνται για να σχηματίσουν ομάδες.
- Οι εγγραφές που έχουν την ίδια τιμή σε όλα τα χαρακτηριστικά του όρου group by τοποθετούνται σε μία ομάδα.
- Τα χαρακτηριστικά στον όρο select πρέπει να είναι μεταξύ αυτών που εμφανίζονται οι **συναθροιστικές συναρτήσεις** ή εμφανίζονται στον όρο **group by**.

Συναθροιστικές Συναρτήσεις (Aggregate Functions)

Ομαδοποίηση Εγγραφών (group by) [2/2]



- ✓ Βρείτε το Μ.Ο. των μισθών κάθε τμήματος.

```
SELECT dept_name, AVG (salary) AS avg_salary  
FROM instructor  
GROUP BY dept_name;
```

- ✓ Βρείτε τον αριθμό των καθηγητών σε κάθε τμήμα που δίδαξαν ένα μάθημα την Άνοιξη του 2010.

```
SELECT dept_name, COUNT (DISTINCT ID) AS instr_count  
FROM instructor NATURAL JOIN teaches  
WHERE semester = 'Spring' AND year = 2010  
GROUP BY dept_name;
```

- Χρησιμοποιείται όταν θέλουμε η συνθήκη να εφαρμόζεται σε **ομάδες** & όχι σε εγγραφές.
- Οι συνθήκες εφαρμόζονται μετά το σχηματισμό των ομάδων (η SQL εφαρμόζει κατηγορήματα στον όρο having, αφού σχηματιστούν οι ομάδες).
- Περιλαμβάνει πάντα συναθροιστικές συναρτήσεις.

- ✓ Βρείτε τα τμήματα όπου ο Μ.Ο. των μισθών των καθηγητών να είναι μεγαλύτερος από 42000.

```
SELECT dept_name, AVG (salary) AS avg_salary  
FROM instructor  
GROUP BY dept_name  
HAVING AVG (salary) > 42000;
```

Συναθροιστικές Συναρτήσεις (Aggregate Functions)

Παραδείγματα “CountryLanguage” [1/2]



DataStories

```
SELECT COUNT (DISTINCT(CountryCode))  
FROM City;
```

```
SELECT MAX (population)  
FROM City;
```

```
SELECT SUM (population) AS Total, District  
FROM City  
GROUP BY District  
ORDER BY Total DESC;
```

Συναθροιστικές Συναρτήσεις (Aggregate Functions)

Παραδείγματα “CountryLanguage” [2/2]



```
SELECT SUM (Population) AS Total, District  
FROM City  
GROUP BY District  
ORDER BY Total DESC LIMIT 15;
```

```
SELECT SUM (Population) AS Total  
FROM City  
HAVING SUM (Population) > 20000;
```

Εμφωλευμένα Ερωτήματα (Nested Queries)

- **select - from - where** έκφραση, που είναι ένθετη μέσα σε ένα άλλο ερώτημα.
- **Χρήσεις:**
 - ▶ Εκτέλεση ελέγχων αν κάποιο στοιχείο ανήκει σ' ένα σύνολο.
 - ▶ Συγκρίσεις συνόλων & προσδιορισμός της τάξης ενός συνόλου (ένθετα υπο-ερωτήματα στον όρο where).
 - ▶ Ένθεση υπο-ερωτημάτων στον όρο from.

Εμφωλευμένα Ερωτήματα (Nested Queries)

Μέλος Ενός Συνόλου [1/2]



- ✓ Βρείτε όλα τα μαθήματα που διδάχθηκαν το Φθινόπωρο του 2009 & την Άνοιξη του 2010.
- ➔ Αρχικά, βρίσκουμε όλα τα μαθήματα που διδάχθηκαν την Άνοιξη του 2010:

```
(SELECT course_id  
FROM section  
WHERE semester = 'Spring' AND year = '2010')
```

Εμφωλευμένα Ερωτήματα (Nested Queries)

Σύγκριση Συνόλων



- ➔ Κατόπιν, βρίσκουμε εκείνα τα μαθήματα που διδάχθηκαν το Φθινόπωρο του 2009 & που εμφανίζονται στο σύνολο των μαθημάτων που λαμβάνονται από το υπο-ερώτημα:

```
SELECT DISTINCT course_id
FROM section
WHERE semester = 'Fall' AND year = 2009 AND
course_id IN (SELECT course_id
               FROM section
               WHERE semester = 'Spring' AND year = 2010);
```


Εμφωλευμένα Ερωτήματα (Nested Queries)

Μέλος Ενός Συνόλου [2/2]



- ✓ Βρείτε τα ονόματα όλων των καθηγητών των οποίων ο μισθός είναι μεγαλύτερος από τουλάχιστον έναν καθηγητή του τμήματος Βιολογίας.

```
SELECT name
FROM instructor
WHERE salary > SOME (SELECT salary
                        FROM instructor
                        WHERE dept_name = 'Biology');
```

Εμφωλευμένα Ερωτήματα (Nested Queries)

Παράδειγμα “CountryLanguage”



DataStories

```
SELECT Name
FROM City
WHERE CountryCode IN (SELECT Code
                       FROM Country
                       WHERE IndepYear > 1920 AND Continent = 'Asia');
```

Outline



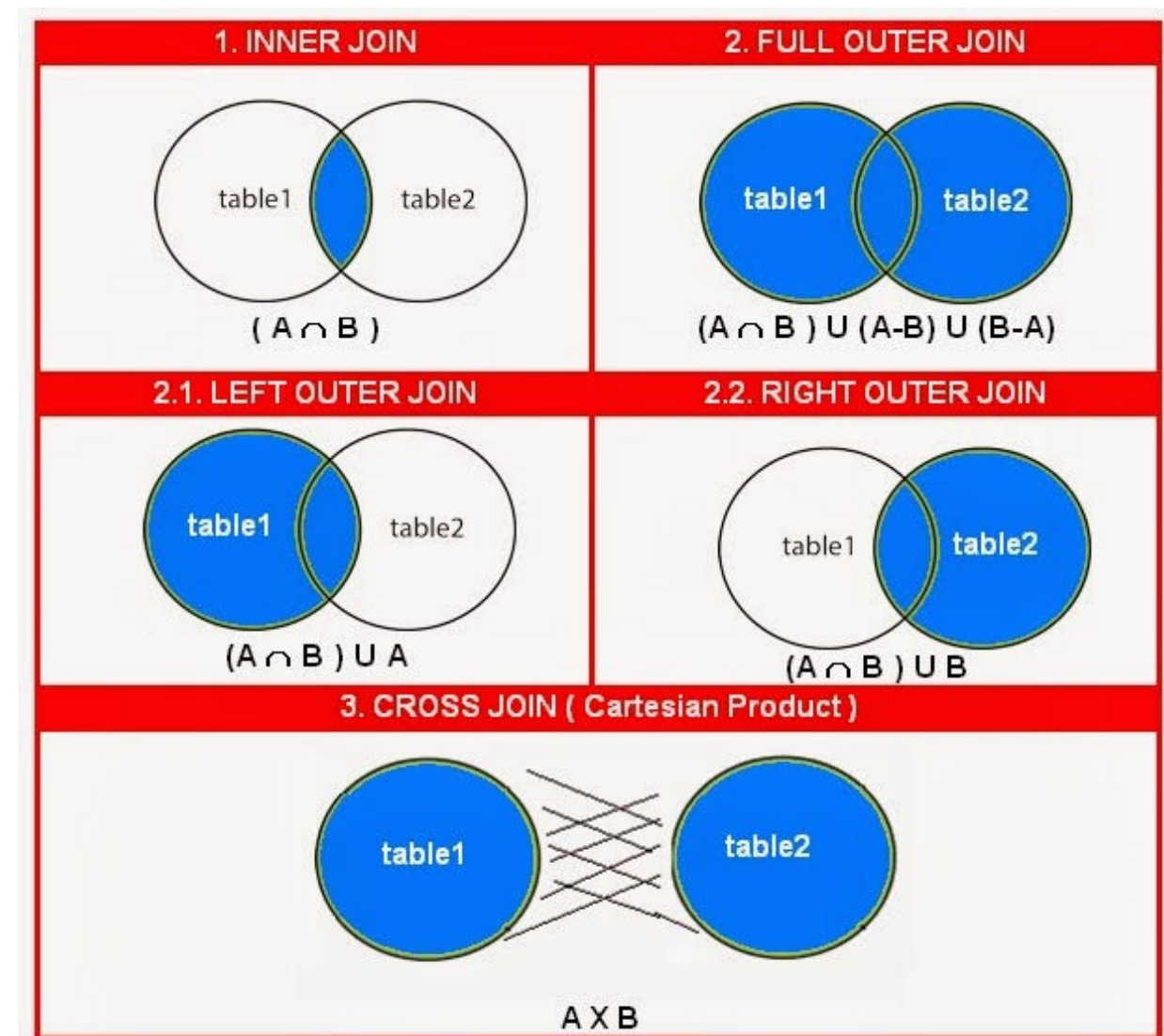
DataStories

- *Hands on (DB)
- Composite Types, Natural Join
- Rename, Like, Order by
- Union, Intersect, Except
- Aggregate Functions, Nested Queries
- **Joins Expressions**
- *Hands on (Queries)



Join Expressions Types

- **CROSS JOIN** (without connection condition)
- **Qualified joins** (with connection condition)
 - ▶ **INNER JOIN**
 - ▶ **OUTER JOIN** (LEFT, RIGHT, FULL)



Join Expressions

Example



DataStories

```
CREATE TABLE T1 (  
  num integer NOT NULL,  
  name text NOT NULL  
);
```

```
CREATE TABLE T2 (  
  num integer NOT NULL,  
  value text NOT NULL  
);
```

```
INSERT INTO T1(num, name) VALUES (1, 'a'), (2, 'b'), (3, 'c');
```

	num integer	name text
1	1	a
2	2	b
3	3	c

```
INSERT INTO T2(num, value) VALUES (1, 'xxx'), (3, 'yyy'), (5, 'zzz');
```

	num integer	value text
1	1	xxx
2	3	yyy
3	5	zzz

Join Expressions

CROSS JOIN

- Καρτεσιανό γινόμενο 2 πινάκων.
- Για κάθε πιθανό συνδυασμό των γραμμών του T1 με το T2, κάθε γραμμή του αποτελέσματος θα αποτελείται από όλες τις στήλες του T1 ακολουθούμενες από όλες τις στήλες του T2.

(T1 \rightarrow n rows, T2 \rightarrow m rows)
then
(T1 **CROSS JOIN** T2 \rightarrow n * m rows)

```
SELECT *  
FROM T1 CROSS JOIN T2;
```

	num integer	name text	num integer	value text
1	1	a	1	xxx
2	1	a	3	yyy
3	1	a	5	zzz
4	2	b	1	xxx
5	2	b	3	yyy
6	2	b	5	zzz
7	3	c	1	xxx
8	3	c	3	yyy
9	3	c	5	zzz

Join Expressions

INNER JOIN

- INNER JOIN ... ON
- Για κάθε γραμμή R1 του T1, το αποτέλεσμα περιέχει μία γραμμή για κάθε γραμμή του T2, η οποία ικανοποιεί τη συνθήκη σύνδεσης συγκρινόμενη με την R1.

```
SELECT *  
FROM T1 INNER JOIN T2 ON T1.num = T2.num;
```

	num integer	name text	num integer	value text
1	1	a	1	xxx
2	3	c	3	yyy

Join Expressions

LEFT OUTER

- LEFT OUTER JOIN ... ON
- Πρώτα εκτελείται ένα INNER JOIN.
- Μετά, για κάθε γραμμή του T1, η οποία δεν ικανοποιεί τη συνθήκη σύνδεσης συγκρινόμενη με οποιαδήποτε γραμμή του T2, προστίθεται στο αποτέλεσμα μία γραμμή με null τιμές στις στήλες του T2.

```
SELECT *  
FROM T1 LEFT JOIN T2 ON T1.num = T2.num;
```

	num integer	name text	num integer	value text
1	1	a	1	xxx
2	2	b		
3	3	c	3	yyy

Join Expressions

RIGHT OUTER

- RIGHT OUTER JOIN ... ON
- Πρώτα εκτελείται ένα INNER JOIN.
- Μετά, για κάθε γραμμή του T2, η οποία δεν ικανοποιεί τη συνθήκη σύνδεσης συγκρινόμενη με οποιαδήποτε γραμμή του T1, προστίθεται στο αποτέλεσμα μία γραμμή με null τιμές στις στήλες του T1.

```
SELECT *  
FROM T1 RIGHT JOIN T2 ON T1.num = T2.num;
```

	num integer	name text	num integer	value text
1	1	a	1	xxx
2	3	c	3	yyy
3			5	zzz

Join Expressions

FULL OUTER JOIN



DataStories

- FULL OUTER JOIN ... ON
- Πρώτα εκτελείται ένα INNER JOIN.
- Μετά, για κάθε γραμμή του T1, η οποία δεν ικανοποιεί τη συνθήκη σύνδεσης συγκρινόμενη με οποιαδήποτε γραμμή του T2, προστίθεται στο αποτέλεσμα μία γραμμή με null τιμές στις στήλες του T2.
- Επίσης, για κάθε γραμμή του T2, η οποία δεν ικανοποιεί τη συνθήκη σύνδεσης συγκρινόμενη με οποιαδήποτε γραμμή του T1, προστίθεται στο αποτέλεσμα μία γραμμή με null τιμές στις στήλες του T1.

```
SELECT *  
FROM T1 FULL JOIN T2 ON T1.num = T2.num;
```

	num integer	name text	num integer	value text
1	1	a	1	xxx
2	2	b		
3	3	c	3	yyy
4			5	zzz

Join Expressions

Προτεραιότητα Συνθηκών

- Η συνθήκη/συνθήκες που θέτουμε στο ON της σύνδεσης, επεξεργάζονται πριν την εκτέλεσή της.
- Η συνθήκη/συνθήκες που θέτουμε στο WHERE, σε ένα ερώτημα που περιέχει σύνδεση με ON, επεξεργάζονται μετά την εκτέλεση της σύνδεσης.

SELECT *

FROM T1 LEFT JOIN T2 ON T1.num = T2.num AND T2.name = 'xxx';

<input type="checkbox"/>	1	a	1	xxx
<input type="checkbox"/>	2	b	[null]	[null]
<input type="checkbox"/>	3	c	[null]	[null]

SELECT *

FROM T1 LEFT JOIN T2 ON T1.num = T2.num

WHERE T2.name = 'xxx';

<input type="checkbox"/>	1	a	1	xxx
--------------------------	---	---	---	-----

Outline



DataStories

- *Hands on (DB)
- Composite Types, Natural Join
- Rename, Like, Order by
- Union, Intersect, Except
- Aggregate Functions, Nested Queries
- Joins Expressions
- ***Hands on (Queries)**



*Hands on

Επαναληπτικές Ασκήσεις [1/8]



[City (ID, Name, CountryCode, District, Population)]

- ✓ Ποιες είναι οι πόλεις; Να εμφανίζονται όλα τα χαρακτηριστικά.
- ✓ Ποιες είναι οι πόλεις; Να εμφανίζεται το όνομα.
- ✓ Ποιες είναι οι πόλεις; Να εμφανίζεται το όνομα, ο πληθυσμός & η περιοχή της πόλης.
- ✓ Ποιες είναι οι πόλεις; Να εμφανίζονται όλα τα χαρακτηριστικά των πόλεων (με ρητή δήλωση).
- ✓ Ποιες είναι οι πόλεις; Να εμφανίζονται όλα τα χαρακτηριστικά, ταξινομημένα κατά όνομα (αύξουσα σειρά).

*Hands on

Επαναληπτικές Ασκήσεις [2/8]



City (ID, Name, CountryCode, District, Population)

- ✓ Ποιες είναι οι πόλεις; Να εμφανίζεται το όνομα & ο πληθυσμός. Η ταξινόμηση να είναι κατά τον πληθυσμό (φθίνουσα σειρά).
- ✓ Ποιες είναι οι πόλεις των οποίων το όνομα ξεκινά από το γράμμα Α; Να εμφανίζονται όλα τα χαρακτηριστικά.
- ✓ Ποιες είναι οι πόλεις των οποίων το όνομα περιέχει το γράμμα α ; Να εμφανίζονται όλα τα χαρακτηριστικά.
- ✓ Ποιες είναι οι πόλεις των οποίων το όνομα περιέχει δύο συνεχόμενα γράμματα α;
- ✓ Ποιες είναι οι πόλεις των οποίων το όνομα έχει στο δεύτερο γράμμα το γράμμα α;

*Hands on

Επαναληπτικές Ασκήσεις [3/8]



City (ID, Name, CountryCode, District, Population)

- ✓ Ποιες είναι οι πόλεις με πληθυσμό μεγαλύτερο από 100 K; Να εμφανίζονται όλα τα χαρακτηριστικά.
- ✓ Ποιες είναι οι πόλεις με πληθυσμό μικρότερο από 2 M; Να εμφανίζονται όλα τα χαρακτηριστικά.
- ✓ Ποιες είναι οι πόλεις με πληθυσμό από 1 M έως 1.5 M; Να εμφανίζονται όλα τα χαρακτηριστικά.
- ✓ Ποιες είναι οι πόλεις με πληθυσμό μεγαλύτερο από 100 K & το όνομά τους ξεκινά με το γράμμα B;

*Hands on

Επαναληπτικές Ασκήσεις [4/8]



City (ID, Name, CountryCode, District, Population)

- ✓ Ποιες είναι οι πόλεις με πληθυσμό μεγαλύτερο από 100 K & το όνομά τους ξεκινά με το γράμμα B & η επαρχία που ανήκουν ξεκινά από το γράμμα Q;
- ✓ Ποιες είναι οι πόλεις οι οποίες έχουν πληθυσμό μεγαλύτερο από 1 M ή το όνομά τους ξεκινά από το γράμμα C;
- ✓ Ποιες είναι οι πόλεις οι οποίες είτε έχουν πληθυσμό μεγαλύτερο από 1 M & το όνομά τους ξεκινά από το γράμμα D, είτε έχουν πληθυσμό μικρότερο από 500 K & το όνομά τους ξεκινά από το γράμμα F; Να ταξινομηθούν κατά όνομα (αύξουσα σειρά).

*Hands on

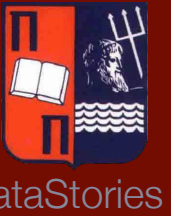
Επαναληπτικές Ασκήσεις [5/8]

City (ID, Name, CountryCode, District, Population)

- ✓ Ποιες είναι οι περιοχές;
- ✓ Ποιοι είναι οι κωδικοί των χωρών;
- ✓ Ποιος είναι ο μεγαλύτερος πληθυσμός από όλες τις πόλεις;
- ✓ Ποιος είναι ο μικρότερος πληθυσμός από όλες τις πόλεις;
- ✓ Ποιος είναι ο μέσος πληθυσμός όλων των πόλεων;
- ✓ Ποιο είναι το άθροισμα του πληθυσμού όλων των πόλεων;
- ✓ Ποιο είναι το άθροισμα του πληθυσμού ανά περιοχή; Να εμφανίζεται το άθροισμα (sum_population) & η περιοχή.

*Hands on

Επαναληπτικές Ασκήσεις [6/8]



City (ID, Name, CountryCode, District, Population)

- ✓ Ποιος είναι ο μέγιστος πληθυσμός ανά περιοχή; Να εμφανίζεται το όνομα της πόλης, το μέγιστο (max_population) & η περιοχή;
- ✓ Ποιος είναι το άθροισμα του πληθυσμού ανά χώρα; Να εμφανίζεται το άθροισμα (sum_population) & η χώρα.

*Hands on

Επαναληπτικές Ασκήσεις [7/8]



DataStories

Country (Code, Name, Continent, Region, SurfaceArea, IndepYear, Population, LifeExpectancy, GNP, GNPOld, LocalName, GovernmentForm, HeadOfState, Capital, Code2)

CountryLanguage (CountryCode, Language, IsOfficial, Percentage)

- ✓ Ποιες είναι οι ανεπίσημες γλώσσες των χωρών; Να εμφανίζεται η χώρα, η γλώσσα & το ποσοστό της γλώσσας;
- ✓ Ποια είναι η χώρα που έχει το μεγαλύτερο ποσοστό που μιλούν ανεπίσημα κάποια γλώσσα;

*Hands on

Επαναληπτικές Ασκήσεις [8/8]



City (ID, Name, CountryCode, District, Population)

CountryLanguage (CountryCode, Language, IsOfficial, Percentage)

- ✓ Ποιες γλώσσες ομιλούνται σε κάθε πόλη; Υποθέτουμε ότι η κατανομή του ποσοστού στο συνολικό πληθυσμό ισχύει για κάθε μέρος του πληθυσμού, άρα & στον πληθυσμό της πόλης. Να εμφανίζεται η πόλη & η γλώσσα.
- ✓ Ποιες γλώσσες ομιλούνται σε κάθε πόλη; Υποθέτουμε ότι η κατανομή του ποσοστού στο συνολικό πληθυσμό ισχύει για κάθε μέρος του πληθυσμού, άρα & στον πληθυσμό της πόλης. Να εμφανίζεται η χώρα, η πόλη & η γλώσσα.

*Hands on

Απαντήσεις [1/7]



DataStories

```
SELECT *  
FROM City;
```

```
SELECT Name  
FROM City;
```

```
SELECT Name, Population, District  
FROM City;
```

```
SELECT CountryCode, Name, Population,  
District FROM City;
```

```
SELECT *  
FROM City  
ORDER BY Name ASC;
```

```
SELECT Name, population  
FROM City  
ORDER BY population DESC;
```

*Hands on

Απαντήσεις [2/7]



DataStories

```
SELECT *  
FROM City  
WHERE Name LIKE 'A%';
```

```
SELECT *  
FROM City  
WHERE Name LIKE 'A%';
```

```
SELECT *  
FROM City  
WHERE Name LIKE '%a%';
```

```
SELECT *  
FROM City  
WHERE Name LIKE '%aa%';
```

```
SELECT *  
FROM City  
WHERE Name LIKE '_a%';
```

*Hands on

Απαντήσεις [3/7]



DataStories

```
SELECT *  
FROM City  
WHERE Name LIKE '%aa%';
```

```
SELECT *  
FROM City  
WHERE Name LIKE '_a%';
```

```
SELECT *  
FROM City  
WHERE Population > 1e5;
```

```
SELECT *  
FROM City  
WHERE Population < 2e6;
```

```
SELECT *  
FROM City  
WHERE Population < 1.5e6 AND Population > 1e6;
```

*Hands on

Απαντήσεις [4/7]

```
SELECT *  
FROM City  
WHERE Population > 1e6 AND Name LIKE 'B%';
```

```
SELECT *  
FROM City  
WHERE Population > 1e6 AND Name LIKE 'B%' AND District LIKE 'Q%';
```

```
SELECT *  
FROM City  
WHERE Population > 1e6 OR Name LIKE 'C%';
```

```
SELECT *  
FROM City  
WHERE (Population > 1e6 AND Name LIKE 'D%') OR (Population < 1e5 AND  
Name LIKE 'F%');
```


*Hands on

Απαντήσεις [5/7]



DataStories

```
SELECT DISTINCT (District)
FROM City;
```

```
SELECT DISTINCT (CountryCode)
FROM City;
```

```
SELECT MAX (Population)
FROM City;
```

```
SELECT MIN (Population)
FROM City;
```

```
SELECT AVG (Population)
FROM City;
```

```
SELECT SUM (Population)
FROM City;
```

*Hands on

Απαντήσεις [6/7]



DataStories

```
SELECT SUM (Population) AS sum_population, District FROM City  
GROUP BY District;
```

```
SELECT MAX (Population) AS max_population, District  
FROM City  
GROUP BY District;
```

```
SELECT SUM (Population) AS sum_population, CountryCode  
FROM City  
GROUP BY CountryCode;
```

```
SELECT Country.Name, CountryLanguage.Language,  
CountryLanguage.Percentage  
FROM Country INNER JOIN CountryLanguage ON  
Country.Code = CountryLanguage.CountryCode  
WHERE CountryLanguage.IsOfficial = 'F'  
ORDER BY Country.Name;
```

*Hands on

Απαντήσεις [7/7]



DataStories

```
SELECT Country.Name, CountryLanguage.percentage, isOfficial
FROM Country INNER JOIN CountryLanguage ON
CountryLanguage.countrycode = Country.code
WHERE (CountryLanguage.percentage =
(SELECT MAX (CountryLanguage.percentage) FROM
CountryLanguage)) AND CountryLanguage.isOfficial='F';
```

```
SELECT City.Name, CountryLanguage.Language
FROM City INNER JOIN CountryLanguage ON
City.CountryCode = CountryLanguage.CountryCode
ORDER BY City.CountryCode;
```

```
SELECT Country.Name, City.Name, CountryLanguage.Language
FROM City INNER JOIN CountryLanguage ON
City.CountryCode = CountryLanguage.CountryCode INNER JOIN
Country ON City.CountryCode = Country.Code
ORDER BY City.CountryCode;
```

Περαιτέρω Μελέτη

Ασκήσεις, Παραδείγματα



- ✓ <http://www.postgresql.org/docs/8.0/static/tutorial-sql.html>
- ✓ www.tutorialspoint.com/postgresql/
- ✓ <https://pgexercises.com/>

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2015), Συστήματα Βάσεων Δεδομένων, Γ' Έκδοση. Εκδόσεις Μόσχος Γκιούρδας.
- The PostgreSQL Global Development Group, PostgreSQL 9.6rc1 Documentation. 2016.
- <https://www.postgresql.org/docs/9.5/static/index.html>
- <https://www.postgresql.org/docs/9.1/static/explicit-locking.html>