

Μαθητεία ΕΠΑΛ
Σημειώσεις Εργαστηριακού Μαθήματος

Ενότητα
«Βάσεις Δεδομένων»

Databases

Βάσεις Δεδομένων και ΣΔΒΔ (DBMS);

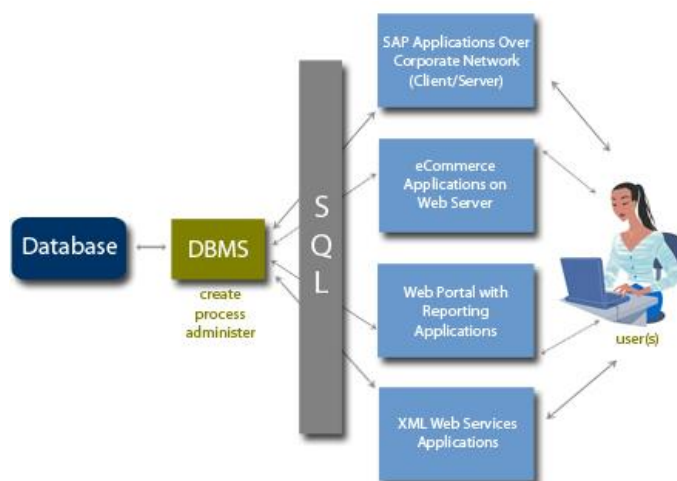
Μια **Βάση Δεδομένων** ή **ΒΔ** (Database) θα μπορούσε να χαρακτηριστεί ως ένα σύνολο δεδομένων/αρχείων τα οποία διαθέτουν υψηλό βαθμό οργάνωσης και είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς χρήστες.



Τα δεδομένα/αρχεία στις ΒΔ δεν δημιουργούνται, ούτε ενημερώνονται από ανεξάρτητες εφαρμογές λογισμικού, αλλά από ένα εξειδικευμένο λογισμικό / σύστημα. Το σύστημα αυτό μεσολαβεί ανάμεσα στα αρχεία δεδομένων και στις εφαρμογές που χρησιμοποιούν οι χρήστες και το ονομάζουμε **Σύστημα Διαχείρισης Βάσης Δεδομένων** ή **ΣΔΒΔ** (Data Base Management System - DBMS).

Το ΣΔΒΔ είναι μια εφαρμογή (database software) που σκοπό έχει το χειρισμό μιας ή περισσότερων ΒΔ, όσον αφορά τη δημιουργία, συντήρηση, επεξεργασία στοιχείων, ελέγχους ασφαλείας κτλ., και την εξυπηρέτηση των χρηστών, όσον αφορά την παροχή στοιχείων και πληροφοριών από την ΒΔ, χωρίς να χρειάζεται να γνωρίζουν με ποιο τρόπο και πού είναι αποθηκευμένα και οργανωμένα αυτά τα δεδομένα.

Ένα ΣΔΒΔ είναι στην ουσία ένας "ενδιάμεσος" μεταξύ του χρήστη και της βάσης δεδομένων και μόνο μέσω αυτού ο χρήστης μπορεί να καταχωρήσει και να αντλήσει πληροφορίες από τη ΒΔ. Μπορεί το ΣΔΒΔ να είναι εγκατεστημένο σ' ένα Η/Υ και να χρησιμοποιείται από μόνο έναν χρήστη (single user system) ή να είναι εγκατεστημένο σε κάποιον κεντρικό εξυπηρετητή (database server) ο οποίος να επικοινωνεί μέσω κάποιου (τοπικού ή απομακρυσμένου) δικτύου και να χρησιμοποιείται από πολλούς χρήστες (multi-user system).

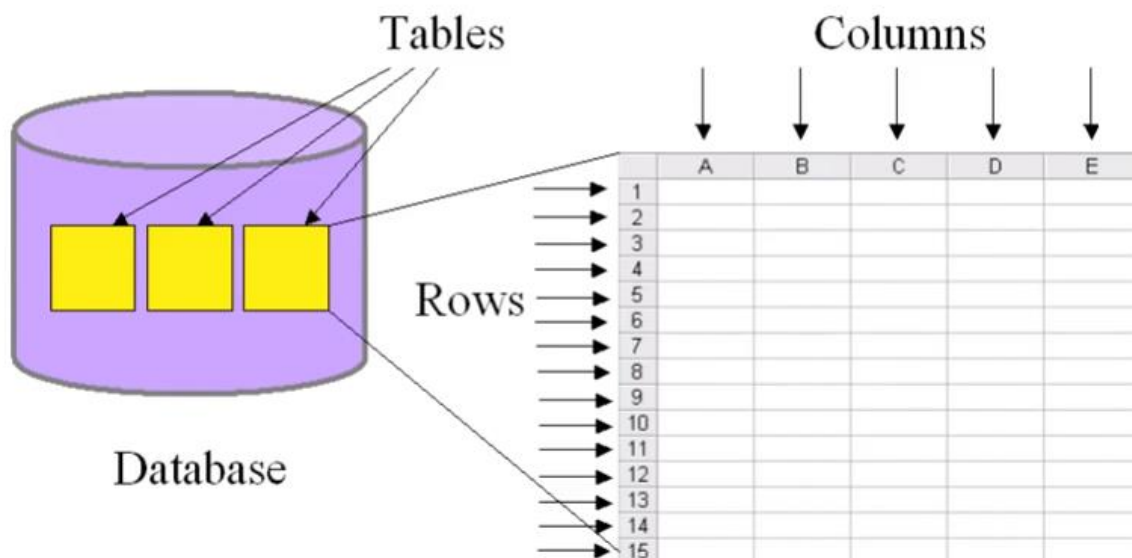


Σχεσιακές Βάσεις Δεδομένων (RDBMS)

Οι **Σχεσιακές ΒΔ** (Relational Databases) είναι οι πιο δημοφιλείς και πιο ευρέως διαδεδομένες. Τα συστήματα DBMS αντίστοιχα ονομάζονται "Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων" (Relational Database Management System - RDBMS) χαρακτηρίζονται γενικότερα από την απλότητα και την ευκολία στην χρήση τους.

Πως οργανώνονται τα δεδομένα στις Σχεσιακές Βάσεις Δεδομένων;

Η οργάνωση των δεδομένων γίνεται σε Πίνακες (Database Tables). Κάθε πίνακας έχει ένα μοναδικό όνομα και απαρτίζεται από **Πεδία (Fields)** που είναι οι **στήλες** και από **Εγγραφές (Records)** που είναι οι **γραμμές**!



Παραδείγματα RDBMS

Τα πιο γνωστά σχεσιακά συστήματα ΒΔ είναι τα παρακάτω:

MySQL, Oracle, SQLite, IBM DB2, PostgreSQL, MS-SQL, MS Access (όπου MS: Microsoft).



Τι είναι η SQL;

Η SQL (Structured Query Language) ή "Δομημένη Γλώσσα Ερωτημάτων" είναι μια **γλώσσα 4ης γενιάς** με την οποία ο χρήστης επικοινωνεί με ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS - Database Management System) για να εκτελέσει διάφορες ενέργειες όπως για παράδειγμα, ανάκτηση δεδομένων, εγγραφή, διαγραφή ή ενημέρωση δεδομένων, κ.α.

Τύποι Δεδομένων στην SQL

Οι βασικοί τύποι δεδομένων που υποστηρίζονται από τα περισσότερα συστήματα είναι οι παρακάτω. Ωστόσο, χρειάζεται να σημειώσουμε ότι ενδέχεται κάποιοι τύποι δεδομένων να διαφέρουν από σύστημα DBMS και γι' αυτό καλύτερα να εξετάζονται στο σύστημα στο οποίο θα γίνει η αποθήκευση των δεδομένων.

- char (n) String σταθερού μήκους n χαρακτήρων
- varchar(n) String μεταβλητού μήκους χαρακτήρων με μέγιστο αριθμό n
- text String για μεγάλου μεγέθους κείμενα από 0 έως 65536 χαρακτήρες
- tinyint Ακέραιος -128 έως 127 (ή 255 ως UNSIGNED)
- smallint Ακέραιος -32.768 έως 32.767 (ή 65535 ως UNSIGNED)
- mediumint Ακέραιος -8.388.608 έως 8.388.607 (ή 16.777.215 ως UNSIGNED)
- int Ακέραιος -2.147.483.648 έως 2.147.483.647 (ή 4.294.967.295 ως UNSIGNED)
- bigint Ακέραιος -9.223.372.036.854.775.808 έως 9.223.372.036.854.775.807 (ή 18.446.744.073.709.551.615 ως UNSIGNED)
- float Πραγματικός Αριθμός
- double Πραγματικός Αριθμός πολύ μεγάλης ακρίβειας
- decimal Πραγματικός Αριθμός στον οποίο μπορούμε να καθορίσουμε το πλήθος των δεκαδικών ψηφίων π.χ. 2 ψηφία
- date Ημερομηνία σε μορφή YYYY-MM-DD
- datetime Ημερομηνία και Ώρα σε μορφή YYYY-MM-DD HH:MI:SS
- timestamp Τρέχουσα Ημ/νία και Ώρα σε μορφή YYYY-MM-DD HH:MI:SS
- time Ώρα σε μορφή HH:MI:SS
- year Έτος σε μορφή 2 ψηφίων (YY) ή 4 ψηφίων (YYYY)

Διερεύνηση! Ποια είναι η διαφορά ανάμεσα στους *Signed* και *Unsigned* τύπους δεδομένων;

Για Περισσότερα Πληροφορίες:

https://www.w3schools.com/sql/sql_datatypes.asp

Οι Εντολές της SQL

Οι πιο σημαντικές εντολές SQL είναι οι παρακάτω:

- **SELECT** - εξαγωγή/εμφάνιση δεδομένων από έναν ή περισσότερους πίνακες
- **INSERT INTO** - εισαγωγή/καταχώρηση δεδομένων
- **UPDATE** - ενημέρωση δεδομένων (εγγραφών σ' έναν πίνακα)
- **DELETE** - διαγραφή δεδομένων (εγγραφών απ' έναν πίνακα)
- **CREATE DATABASE** - δημιουργία μιας νέας ΒΔ
- **ALTER DATABASE** - τροποποίηση μιας ΒΔ
- **CREATE TABLE** - δημιουργία νέου πίνακα
- **ALTER TABLE** - τροποποίηση ενός πίνακα (π.χ. για την προσθήκη ενός νέου πεδίου)
- **DROP TABLE** - διαγραφή ενός πίνακα
- **CREATE INDEX** - δημιουργία ενός νέου ευρετηρίου (index)
- **DROP INDEX** - διαγραφή ενός ευρετηρίου (index)

Παραδείγματα Εντολών SQL "SELECT"

Εμφάνιση όλων των εγγραφών ενός πίνακα

Σύνταξη: **SELECT * FROM <table>;**
Παράδειγμα: **SELECT * FROM Employees;**

Εμφάνιση συγκεκριμένων στηλών

SELECT lastname, age, email FROM Employees;
SELECT email, firstname, lastname, age FROM Employees;

Φιλτράρισμα Εγγραφών ενός Πίνακα

SELECT email FROM Employees WHERE lastname = 'ΠΑΠΑΔΟΠΟΥΛΟΣ';
SELECT email FROM Employees WHERE lastname LIKE 'ΠΑΠ%';
SELECT email FROM Employees WHERE lastname LIKE '%ΠΑΠ%';
SELECT email FROM Employees WHERE age > 30;

Διερεύνηση! Ποια είναι η διαφορά ανάμεσα στο 2ο και το 3ο ερώτημα;

Ταξινόμηση των Αποτελεσμάτων

```
SELECT lastname, age, email FROM Employees ORDER BY lastname;  
SELECT lastname, age, email FROM Employees ORDER BY email ASC;  
SELECT lastname, age, email FROM Employees ORDER BY email DESC;  
SELECT lastname, age, email FROM Employees ORDER BY age, lastname DESC;
```

Παραδείγματα Εντολών SQL "INSERT"

Εισαγωγή μιας νέας εγγραφής σ' έναν Πίνακα

```
INSERT INTO Employees (lastname, firstname, age, email) VALUES ('Παπαδόπουλος', 'Κώστας', 35,  
'papkostas@gmail.com');
```

Διερεύνηση! **α)** Αν δεν συμπεριλάβουμε στην εντολή το πρωτεύον κλειδί (*primary key*) του πίνακα, θα αντιμετωπίσουμε κάποιο πρόβλημα και τότε; **β)** Υπάρχει δυνατότητα να μην συμπεριλάβουμε κάποιο πεδίο ή πεδία του πίνακα στην εντολή *INSERT*; Αν Ναι, τι συμβαίνει σ' αυτή την περίπτωση, τι δεδομένα θα καταχωρηθούν; **γ)** Σε ποια πεδία βάζουμε τα μονά εισαγωγικά και τότε δεν απαιτούνται; **δ)** Πώς πραγματοποιείται η εισαγωγή μιας ημερομηνία (πεδίο τύπου *DATE*) και πώς η εισαγωγή ενός στην περίπτωση της Ημερομηνίας και Ώρας (πεδίο τύπου *DATETIME*);

Παραδείγματα Εντολών SQL "UPDATE"

Ενημέρωση μιας εγγραφής ενός Πίνακα χρησιμοποιώντας το Πρωτεύον Κλειδί

```
UPDATE Employees SET age = 21, email = 'john@gmail.com' WHERE id = 5;
```

Μαζική ενημέρωση των εγγραφών ενός Πίνακα

```
UPDATE Employees SET salary = 501.23 WHERE age < 20;
```

```
UPDATE Employees SET email = '';
```

Διερεύνηση! α) Τι αλλάζει στα δεδομένα με την εκτέλεση των παραπάνω εντολών. β) Πόσες εγγραφές θα επηρεαστούν από τις παραπάνω εντολές; γ) Πώς μπορούμε να το γνωρίζουμε πριν την εκτέλεσή τους;

Παραδείγματα Εντολών SQL "DELETE"

Διαγραφή μιας εγγραφής ενός Πίνακα χρησιμοποιώντας το Πρωτεύον Κλειδί

```
DELETE FROM Employees WHERE id = 5;
```

Μαζική διαγραφή πολλών εγγραφών ενός Πίνακα

```
DELETE FROM Employees WHERE age >= 18;
```

Διερεύνηση! α) Πόσες εγγραφές θα διαγραφούν από την παραπάνω εντολή; β) Πώς μπορούμε να το γνωρίζουμε πριν την εκτέλεσή της;

Μαζική διαγραφή ΟΛΩΝ των εγγραφών ενός Πίνακα

```
DELETE FROM Employees;
```

Διερεύνηση! α) Πόσες εγγραφές θα διαγραφούν; β) Σε περίπτωση λάθους, πως μπορούμε να επαναφέρουμε τις εγγραφές του Πίνακα;

Παραδείγματα Εντολών SQL

Δημιουργία μιας νέας Βάσης Δεδομένων - Εντολή SQL: **CREATE DATABASE**

```
CREATE DATABASE demo_db1;
```

Δημιουργία ενός Πίνακα - Εντολή SQL: **CREATE TABLE**

```
CREATE TABLE Employees (  
    id int NOT NULL,  
    lastname varchar(50),  
    firstname varchar(50),  
    age smallint,  
    dob date,  
    email varchar(100),  
    PRIMARY KEY (id)  
);
```

Διαγραφή ενός Πίνακα - Εντολή SQL: **DROP TABLE**

```
DROP TABLE Employees;
```

Τροποποίηση της δομής ενός Πίνακα - Εντολή SQL: **ALTER TABLE**

```
ALTER TABLE Employees ADD COLUMN salary float;  
  
ALTER TABLE Employees DROP COLUMN age;  
  
ALTER TABLE Employees MODIFY id AUTO_INCREMENT;
```

Διερεύνηση! Τι αλλαγές θα πραγματοποιήσουν κάθε μία από τις παραπάνω εντολές;

Κανονικοποίηση (Normalization)

Η **κανονικοποίηση** αποτελεί ένα πολύ σημαντικό ζήτημα στον σχεδιασμό Βάσεων Δεδομένων. Οι πίνακες που προκύπτουν συνήθως μετά από τον αρχικό σχεδιασμό (μοντελοποίηση) περιέχουν πληροφορίες που επαναλαμβάνονται. Η επαναλαμβανόμενη πληροφορία οδηγούν σε πολλαπλές γραμμές εντός του ίδιου πίνακα ή επανάληψη της ίδια πληροφορίας σε διαφορετικούς πίνακες που οδηγεί σε σημαντικά προβλήματα κατά την καταχώρηση, ενημέρωση και διαγραφή των πληροφοριών. Με την εφαρμογή των κανόνων κανονικοποίησης απλοποιούνται οι σχέσεις μεταξύ των δεδομένων με **στόχο την εξάλειψη της επαναληπτικότητας των δεδομένων** όπως για παράδειγμα των διπλών εγγραφών.

Ασκήσεις στις Βάσεις Δεδομένων

Άσκηση #1 - Εμφάνιση πληροφοριών με την εντολή SQL "SELECT"

Επισκεφτείτε στο **W3Schools** την παρακάτω διεύθυνση απ' όπου μπορείτε να εκτελέσετε εντολές SQL (SQL Queries/Statements) σε μια online Βάση Δεδομένων.

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

Δημιουργήστε την κατάλληλη εντολή SQL για να απαντήσετε στις παρακάτω ερωτήσεις:

1. Εμφανίστε τα στοιχεία όλων των πελατών (πίνακας Customers);

ΑΠΑΝΤΗΣΗ: SELECT * FROM Customers;

2. Εμφανίστε όλα τα στοιχεία του πίνακα ταξινομημένα με βάσει την Πόλη (City).

ΑΠΑΝΤΗΣΗ:

3. Εμφανίστε όλα τα στοιχεία των πελατών που είναι από «UK».

ΑΠΑΝΤΗΣΗ:

4. Εμφανίστε όλα τα στοιχεία των πελατών που είναι από το Παρίσι «Paris».

ΑΠΑΝΤΗΣΗ:

5. Εμφανίστε όλα τα στοιχεία των πελατών ταξινομημένα με βάσει το Επώνυμο.

ΑΠΑΝΤΗΣΗ:

6. Εμφανίστε τους πελάτες που η επωνυμία τους (CustomerName) περιέχει την λέξη «Stop».

ΑΠΑΝΤΗΣΗ:

7. Εμφανίστε όλα τα προϊόντα των οποίων η τιμή είναι μεγαλύτερη από 50€.

ΑΠΑΝΤΗΣΗ:

8. Εμφανίστε όλα τα προϊόντα των οποίων η τιμή είναι μεγαλύτερη από 50€ ταξινομημένα πρώτα αυτά με την μεγαλύτερη τιμή και τελευταία αυτά με την μικρότερη τιμή.

ΑΠΑΝΤΗΣΗ:

Ασκήσεις Πιστοποίησης

29. Δραστηριότητα:

1. Δημιουργήστε μία νέα βάση δεδομένων για ταινίες
2. Μέσα στη βάση δημιουργήστε έναν πίνακα με πεδία για τον αύξοντα αριθμό (πρωτεύων κλειδί), τον τίτλο της ταινίας, το έτος κυκλοφορίας της, τη διάρκειά της σε λεπτά, και το αν είναι τρισδιάστατη (1) ή όχι (0).
3. Εισάγετε στον πίνακα την εγγραφή:
1, "The Dark Knight", 2008, 152, 0
4. Τοποθετήστε σε μια λίστα τις ακόλουθες τιμές:
2, "The Godfather", 1972, 175, 0
3, "Amelie", 2001, 122, 0
4, "The Sting", 1973, 129, 0
5, "How to Train Your Dragon", 2010, 98, 1
6, "Avatar", 2009, 162, 1
7, "Gravity", 2013, 90, 1
8, "Toy Story 3", 2010, 103, 1
9, "Cloudy With a Chance of Meatballs", 2009, 90, 0
10, "Casablanca", 1942, 102, 0
5. Εισάγετε τις τιμές αυτές στον πίνακα, χρησιμοποιώντας τη μέθοδο εισαγωγής πολλαπλών εγγραφών.
6. Εμφανίστε τους τίτλους και τη διάρκεια των ταινιών, με σειρά από την πιο πρόσφατη προς την παλαιότερη.
7. Εμφανίστε τους τίτλους των ταινιών που ξεκινούν από "T". Η εμφάνιση να γίνει σε ξεχωριστή γραμμή για κάθε εγγραφή.
8. Εμφανίστε τα πλήρη στοιχεία των ταινιών που είναι τρισδιάστατες και έχουν διάρκεια πάνω από 100 λεπτά.

* Για εξοικονόμηση χρόνου τα δεδομένα μπορούν να δοθούν σε ένα αρχείο κειμένου, ώστε να μην χρειάζεται να πληκτρολογηθούν.

30. Δραστηριότητα:

1. Δημιουργήστε μία νέα βάση δεδομένων για τους υπαλλήλους μιας εταιρείας
2. Μέσα στη βάση δημιουργήστε έναν πίνακα με πεδία για τον αύξοντα αριθμό (πρωτεύων κλειδί), το όνομα του υπαλλήλου, τη θέση στην οποία εργάζεται, την ηλικία του και το μισθό του σε ευρώ.
3. Εισάγετε στον πίνακα την εγγραφή:
1, "Georgiou", "Sales", 25, 800
4. Τοποθετήστε σε μια λίστα τις ακόλουθες τιμές:
2, "Ioannou", "Sales", 850, 28
3, "Kali", "Secretariat", 700, 22
4, "Konti", "Support", 950, 35
5, "Vrettos", "Administration", 1400, 42
6, "Alexiou", "Sales", 850, 30
7, "Kokkinos", "Support", 900, 34
8, "Halatsis", "Secretariat", 1037, 38
9, "Grigoradou", "Administration", 1180, 40
10, "Anastasiou", "Sales", 980, 44
5. Εισάγετε τις τιμές αυτές στον πίνακα, χρησιμοποιώντας τη μέθοδο εισαγωγής πολλαπλών εγγραφών.
6. Αυξήστε κατά 10% το μισθό των υπαλλήλων που εργάζονται στις πωλήσεις (Sales)
7. Εμφανίστε τους υπαλλήλους που το όνομά τους ξεκινά από "K" με σειρά από την μεγαλύτερη προς τη μικρότερη ηλικία. Η εμφάνιση να γίνει σε ξεχωριστή γραμμή για κάθε εγγραφή.
8. Εμφανίστε τα πλήρη στοιχεία των υπαλλήλων που έχουν ηλικία μεγαλύτερη ή ίση των 35 ετών και μισθό μικρότερο των 1000 ευρώ.

* Για εξοικονόμηση χρόνου τα δεδομένα μπορούν να δοθούν σε ένα αρχείο κειμένου, ώστε να μην χρειάζεται να πληκτρολογηθούν.

31. Δραστηριότητα:

1. Δημιουργήστε μία νέα βάση δεδομένων για αυτοκίνητα.
2. Μέσα στη βάση δημιουργήστε έναν πίνακα με πεδία για τον αύξοντα αριθμό (πρωτεύων κλειδί, αυτόματη αρίθμηση), το όνομα μοντέλου, τον κυβισμό, την υποδύναμη και το χρόνο επιτάχυνσης 0-100 (πραγματικός).
3. Εισάγετε στον πίνακα την εγγραφή:
"AUDI A5 1.8 TFSI ",1798,170,7.9
4. Τοποθετήστε σε μια λίστα τις ακόλουθες τιμές:
"FIAT 500 1.2",1242,69,12.9
"CITROEN C3 1.1",1124,60,16.5
"MERCEDES B 180",1595,122,10.4
"KIA RIO 1.2 5D", 1248,85,13.1
"LEXUS IS 250", 2500,208,8.4
"NISSAN MICRA 1.2",1198,80,13.7
"OPEL ANTARA 2.4",2384,167,10.3
"MINI COOPER 1.6",1598,122,9.1
"TOYOTA YARIS 1.33",1329,99,11.7
"SEAT LEON 1.4 TSI", 1395,140,8.2
5. Εισάγετε τις τιμές αυτές στον πίνακα, χρησιμοποιώντας τη μέθοδο εισαγωγής πολλαπλών εγγραφών.
6. Εμφανίστε το όνομα και τον κυβισμό των αυτοκινήτων που έχουν κυβισμό κάτω από 1300, μαζί με αυτά που έχουν κυβισμό πάνω από 2000. Η εμφάνιση να γίνει σε ξεχωριστή γραμμή για κάθε εγγραφή.
7. Εμφανίστε όλα τα στοιχεία των αυτοκινήτων που έχουν κυβισμό μικρότερο του 1600 και υποδύναμη μεγαλύτερη ή ίση του 80.
8. Διαγράψτε τα αυτοκίνητα με επιτάχυνση μεγαλύτερη του 10.

Ερωτήσεις Πιστοποίησης

- 116.** Συμπληρώστε τη λέξη που λείπει.
Οι βάσεις δεδομένων που μπορεί η Python να χειρισθεί με χρήση της βιβλιοθήκης sqlite3 ακολουθούν το _____ μοντέλο;
- α. Σχεσιακό
 - β. Ιεραρχικό
 - γ. Δικτυωτό
 - δ. Κατηγορικό
- 117.** Συμπληρώστε τις λέξεις που λείπουν αντιστοιχίζοντας τους αριθμούς των κενών με τα γράμματα των λέξεων. Δεν θα χρησιμοποιηθούν όλες οι λέξεις.
- Σε μία _____ (1) βάση δεδομένων τα δεδομένα οργανώνονται σε _____ (2). Κάθε γραμμή περιέχει τις πληροφορίες για ένα στοιχείο του πίνακα και αποκαλείται _____ (3). Κάθε στήλη χαρακτηρίζει κάποια ιδιότητα του πίνακα και καλείται _____ (4).
- α. Πεδίο
 - β. Εγγραφή
 - γ. Αντικείμενο
 - δ. Δομές
 - ε. Σχεσιακή
 - στ. Πίνακες
- 118.** Ποια από τις ακόλουθες εντολές πρέπει να χρησιμοποιηθεί για την δημιουργία ενός αντικειμένου χειριστή για τη χρήση της βάσης δεδομένων school.db που βρίσκεται στο φάκελο dbase του δίσκου (D:);
- α. `conn=sqlite3.open('c:/dbase/school.db')`
 - β. `conn=sqlite3.assign('c:/dbase/school.db')`
 - γ. `conn=sqlite3.connect('c:/dbase/school.db')`
 - δ. `conn=sqlite3.access('c:/dbase/school.db')`
- 119.** Έστω μια βάση δεδομένων που έχει έναν πίνακα εργαζομένων με όνομα γπαλ στον οποίο, μεταξύ άλλων, καταχωρείται και ο μισθός του κάθε εργαζόμενου στο πεδίο mis. Με ποια εντολή θα γίνει αύξηση κατά 10% του μισθού των εργαζομένων που αμείβονται με 600 ευρώ ή και λιγότερα;
- α. `c.execute("SELECT mis FROM γπαλ WHERE mis<=600")`
 - β. `c.execute("SELECT mis, mis*1.1 FROM γπαλ WHERE mis<=600")`
 - γ. `c.execute("""MODIFY mis TO mis*1.1 WHERE mis<=600""")`
 - δ. `c.execute("""UPDATE γπαλ SET mis=mis*1.1 WHERE mis<=600""")`

120. Έστω μια βάση δεδομένων που έχει έναν πίνακα με όνομα *ylika*. Με ποια εντολή θα διαγραφεί ο πίνακας;

- α. `c.execute("""DELETE * FROM ylika""")`
- β. `c.execute("""ERASE * FROM ylika""")`
- γ. `c.execute("DROP TABLE ylika")`
- δ. `c.execute("""SELECT * FROM ylika DELETE""")`

121. Έστω μια βάση δεδομένων που έχει έναν πίνακα με όνομα *ylika*. Με ποια εντολή θα διαγραφούν οι εγγραφές του πίνακα που έχουν την τιμή 0 στο πεδίο *apoth*;

- α. `c.execute("""DELETE ROWS FROM ylika WHERE apoth=0""")`
- β. `c.execute("""DROP ROWS FROM ylika WHERE apoth=0""")`
- γ. `c.execute("SELECT DELETE FROM ylika WHERE apoth=0")`
- δ. `c.execute("""DELETE FROM ylika WHERE apoth=0""")`

122. Ποιο από τα ακόλουθα τμήματα κώδικα θα έχει σαν αποτέλεσμα την προβολή στην οθόνη των 5 πρώτων εγγραφών που επιστρέφει ένα ερώτημα SQL;

α. `for i in range(0,5):`
`a = c.fetchone()`
`print a`

β. `for i in range(0,5):`

`a = c.fetchone()`
`print a`

γ. `a = c.fetch(5)`
`print a`

δ. `for i in range(0,5):`
`a = c.fetchall()`
`print a`

123. Αν σε μία βάση δεδομένων υπάρχει ένας πίνακας με όνομα `albums` που περιέχει ένα πεδίο με όνομα `title` με τίτλους cd μουσικής, αντιστοιχίστε τα γράμματα των ακόλουθων εντολών με τους αριθμούς της περιγραφής των δεδομένων που αυτές θα επιστρέψουν.

- α. `c.execute("SELECT title FROM albums WHERE title LIKE 'n%'" )`
- β. `c.execute("SELECT title FROM albums WHERE title NOT LIKE '%n%'" )`
- γ. `c.execute("SELECT title FROM albums WHERE title LIKE '%n%'" )`
- δ. `c.execute("SELECT title FROM albums WHERE title UNLIKE '%n%'" )`
- ε. `c.execute("SELECT title FROM albums WHERE title > 'n'" )`
- στ. `c.execute("SELECT title FROM albums WHERE title LIKE '%n%'" )`

- 1. Τα ονόματα των album που περιέχουν το γράμμα n
- 2. Τα ονόματα των album που ξεκινούν με το γράμμα n
- 3. Τα ονόματα των album που δεν περιέχουν το γράμμα n
- 4. Τα ονόματα των album που τελειώνουν με το γράμμα n

124. Σε μία βάση δεδομένων υπάρχει ένας πίνακας με όνομα `tracks` με πληροφορίες για κομμάτια μουσικής. Ο πίνακας αυτός περιέχει μεταξύ άλλων ένα πεδίο με όνομα `name` για τον τίτλο κάθε κομματιού και ένα πεδίο με όνομα `ms` που είναι η διάρκεια του κομματιού. Με ποια εντολή θα ανακτηθούν τα ονόματα των κομματιών με διάρκεια μεταξύ 180000 και 600000;

- α. `c.execute("SELECT name FROM tracks WHERE ms<180000 and ms>600000")`
- β. `c.execute("SELECT name FROM tracks WHERE ms>180000 and ms>600000")`
- γ. `c.execute("SELECT name FROM tracks WHERE ms>=180000 and ms<=600000")`
- δ. `c.execute("SELECT name FROM tracks WHERE ms<180000 and ms<600000")`

125. Σε μία βάση δεδομένων υπάρχει ένας πίνακας με όνομα `tracks` με πληροφορίες για κομμάτια μουσικής. Ο πίνακας αυτός περιέχει μεταξύ άλλων ένα πεδίο με όνομα `name` για τον τίτλο κάθε κομματιού και ένα πεδίο με όνομα `ms` που είναι η διάρκεια του κομματιού. Με ποια εντολή θα ανακτηθούν τα ονόματα και οι διάρκειες των κομματιών με σειρά από την μικρότερη προς τη μεγαλύτερη διάρκεια;

- α. `c.execute("SELECT name, ms FROM tracks ORDER BY name")`
- β. `c.execute("SELECT name, ms FROM tracks ORDER BY ms")`
- γ. `c.execute("SELECT name, ms FROM tracks ORDER BY name DESC")`
- δ. `c.execute("SELECT name, ms FROM tracks ORDER BY ms DESC ")`

198. Έστω η κενή βάση δεδομένων "test.db". Ποιο θα είναι το αποτέλεσμα της εκτέλεσης των παρακάτω εντολών;

```
import sqlite3
conn=sqlite3.connect('test.db')
c=conn.cursor()
c.execute("""CREATE TABLE t (id INTEGER PRIMARY KEY not null, onoma TEXT)""")
c.execute("INSERT INTO t VALUES (1, 'JONSON')")
c.close()
conn.close()
```

- α. Θα δημιουργηθεί ο πίνακας και θα εισαχθεί σε αυτόν μία εγγραφή.
- β. Θα δημιουργηθεί ο πίνακας, αλλά δεν θα γίνει εισαγωγή δεδομένων.
- γ. Θα εμφανιστεί μήνυμα λάθους κατά την προσπάθεια δημιουργίας του πίνακα.
- δ. Θα εμφανιστεί μήνυμα λάθους κατά την προσπάθεια εισαγωγής της εγγραφής.

199. Με ποια από τις παρακάτω εντολές θα δημιουργηθεί σε μία βάση δεδομένων ένας νέος πίνακας με όνομα countries και πεδία id (ακέραιος, αυτόματη αρίθμηση, πρωτεύων κλειδί), name (κείμενο), area (πραγματικός αριθμός), flag (εικόνα);

- α.

```
c.execute("""CREATE TABLE countries (id INTEGER PRIMARY KEY,
name TEXT, area REAL, flag PICTURE)""")
```
- β.

```
c.execute("""CREATE TABLE countries (id AUTOINCREMENT PRIMARY
KEY, name TEXT, area REAL, flag IMAGE)""")
```
- γ.

```
c.execute("""CREATE TABLE countries (id INTEGER PRIMARY KEY
AUTOINCREMENT, name TEXT, area REAL, flag BLOB)""")
```
- δ.

```
c.execute("""CREATE TABLE countries (id INTEGER PRIMARY KEY,
name TEXT, area REAL AUTOINCREMENT, flag BINARY)""")
```

200. Έστω μια βάση δεδομένων που έχει έναν πίνακα με όνομα pelates με πέντε πεδία. Με ποια από τις παρακάτω εντολές μπορούν να προστεθούν σε αυτόν όσες εγγραφές είναι αποθηκευμένες σε μία λίστα p;

- α.

```
c.executemany("""INSERT INTO pelates VALUES (?, ?, ?, ?, ?)""", p)
```
- β.

```
c.executemany("""INSERT INTO pelates VALUES""", p)
```
- γ.

```
c.executemany("""INSERT INTO pelates VALUES (?, ?, ?, ?, ?)""")
```
- δ.

```
c.executemany("""INSERT INTO pelates VALUES""", p, (?, ?, ?, ?, ?))
```

201. Έστω μια βάση δεδομένων που έχει έναν πίνακα με όνομα `xores` με πεδία `id`, `on`, `ekt`, `plith`. Με ποια από τις παρακάτω εντολές μπορούν να προστεθεί σε αυτόν μία εγγραφή που θα αποτελείται από τις τιμές των μεταβλητών `aa`, `name`, `area`, `ppl`;

- α. `c.execute("""INSERT INTO xores (id,on,ekt,plith) """)`
- β. `c.execute("""INSERT INTO xores (aa,name,area,ppl) """)`
- γ. `c.execute("""INSERT INTO xores(id,on,ekt,plith) VALUES
(?, ?, ?, ?) """, (aa,name,area,ppl))`
- δ. `c.execute("""INSERT INTO xores(aa,name,area,ppl) VALUES
(?, ?, ?, ?) """, (id,on,ekt,plith))`

202. Σε μία βάση δεδομένων υπάρχει ένας πίνακας με όνομα `tracks` με πληροφορίες για κομμάτια μουσικής. Ο πίνακας αυτός περιέχει μεταξύ άλλων ένα πεδίο με όνομα `name` για τον τίτλο κάθε κομματιού και ένα πεδίο με όνομα `ms` που είναι η διάρκεια του κομματιού. Με ποια εντολή θα ανακτηθούν τα ονόματα και οι διάρκειες των κομματιών με διάρκεια < 180000 , μαζί με αυτών που έχουν διάρκεια > 600000 .

- α. `c.execute("SELECT name, ms FROM tracks WHERE ms<180000 and ms>600000")`
- β. `c.execute("SELECT name, ms FROM tracks WHERE ms<180000 and NOT ms>600000")`
- γ. `c.execute("SELECT name, ms FROM tracks WHERE ms BETWEEN 180000 and 600000")`
- δ. `c.execute("SELECT name, ms FROM tracks WHERE ms<180000 or ms>600000")`

203. Σε μία βάση δεδομένων υπάρχει ένας πίνακας με όνομα `tracks` με πληροφορίες για κομμάτια μουσικής. Ο πίνακας αυτός περιέχει μεταξύ άλλων ένα πεδίο με όνομα `name` για τον τίτλο κάθε κομματιού και ένα πεδίο με όνομα `ms` που είναι η διάρκεια του κομματιού. Με ποια εντολή θα ανακτηθούν τα ονόματα και οι διάρκειες των κομματιών για τα κομμάτια με διάρκεια μικρότερη από 130000 με σειρά από την μεγαλύτερη προς τη μικρότερη διάρκεια;

- α. `c.execute("SELECT name, ms FROM tracks WHERE ms<130000 ORDER BY ms DESC")`
- β. `c.execute("SELECT name, ms FROM tracks WHERE ms<> 130000 ORDER BY ms")`
- γ. `c.execute("SELECT name, ms FROM tracks WHERE ms< 130000 ORDER BY ms")`
- δ. `c.execute("SELECT name, ms FROM tracks ORDER BY ms DESC ")`

Βιβλιογραφία - Πηγές

- W3schools - SQL
<https://www.w3schools.com/sql/>
- Tutorials Point: Learn SQL
<https://www.tutorialspoint.com/sql/>
- Python Sqlite API 2.0 Interface (sqlite3)
<https://docs.python.org/3.7/library/sqlite3.html>
- Βιβλίο Γ' Τάξης ΕΠΑΛ - Τομέα Πληροφορικής "Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές τους στο Διαδίκτυο"
http://www.iep.edu.gr/images/IEP/EPISTIMONIKI_YPIRESIA/Epist_Monades/B_Kyklos/Tee/2016/BEpal/2016_BEpal_Syst_diaxeirshshs_BD.pdf
- Βιβλίο Γ' Τάξης ΕΠΑΛ - Τομέα Πληροφορικής "Σχεδιασμός και Ανάπτυξη Διαδικτυακών Εφαρμογών"
http://www.iep.edu.gr/images/IEP/EPISTIMONIKI_YPIRESIA/Epist_Monades/B_Kyklos/Tee/2016/GEpal/2016_GEpal_DesnDev_app.pdf
- ΕΟΠΠΕΠ - Θέματα Εξετάσεων Πιστοποίησης Αποφοίτων Μεταλυκειακού Έτους - Τάξης Μαθητείας των ΕΠΑ.Λ. της ειδικότητας "Τεχνικός Εφαρμογών Πληροφορικής"
https://www.eoppep.gr/images/EPAL/epal_texnikos_plirofo.pdf

Απαντήσεις

116. α

117. $1 \rightarrow \epsilon, 2 \rightarrow \sigma\tau, 3 \rightarrow \beta, 4 \rightarrow \alpha$

118. γ

119. δ

120. γ

121. δ

122. Η απάντηση α είναι ίδια με την β! η απάντηση γ και δ είναι λανθασμένες!

123. $\alpha \rightarrow 2, \beta \rightarrow 3, \gamma \rightarrow 4, \delta \rightarrow \text{Λάθος}, \epsilon \rightarrow \text{Λάθος}, \sigma\tau \rightarrow 1$

124. γ

125. β

198. α

199. γ

200. α

201. γ

202. δ

203. α