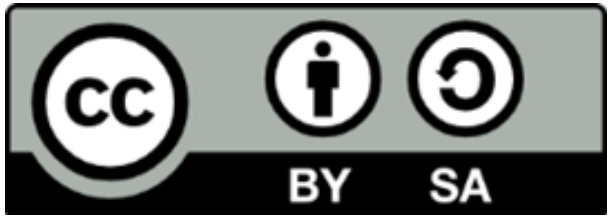
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ Ι

Ενότητα 5: SQL (Απλή SELECT)

Ευαγγελίδης Γεώργιος
Τμήμα Εφαρμοσμένης Πληροφορικής

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

•Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
•Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
•Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοποί ενότητας

προβολή, επιλογή, κανονικές εκφράσεις, γινόμενο, φυσική σύζευξη, συναρτήσεις, ένωση, τομή, διαφορά, διπλότυπα, any, all, υπο-αιτήματα

Query 01

```
-- απλή προβολή πεδίων ενός πίνακα
select pid, name
from performer;
```

Query 02

```
-- συνθήκη επιλογής πάνω σε πίνακα και
-- προβολή, δοκιμάστε και =, >=, <=, >, <, <>
select pid, name
from performer
where pid > 5;
```

Query 03

```
-- συνθήκη συμμετοχής σε σύνολο
select pid, name
from performer
where pid in (3, 4, 9);
```

Query 04

```
-- συνθήκη συμμετοχής σε διάστημα
-- τιμών [α, β]
select pid, name
from performer
where pid between 3 and 8;
```

Query 05

```
-- χρήση κανονικών εκφράσεων (reg exp)
-- ΠΡΟΣΟΧΗ: η Oracle είναι case sensitive
-- στις reg exp
select pid, name
from performer
where name like '%adel%';  -- και '_adel%'
```

Query 06

```
-- προβολή όλων των πεδίων του πίνακα
select *
from cd;
```

Query 07

```
-- αληθής συνθήκη επιλογής (επιστρέφεται
-- ολόκληρος ο πίνακας)
-- δοκιμάστε με 5 = 2
select *
from cd
where 2 = 2;
```

Query 08

```
-- γινόμενο πινάκων
select *
from cd, track;
```

Query 09

```
-- φυσική σύζευξη
select ctitle, pos
from cd, track
where cd.cid = track.cid;
```

Query 10

```
-- στη select χρειαζόμαστε performer.pid ή
-- track.pid, αλλιώς υπάρχει ασάφεια για το ποιο
-- pid μιλάμε
-- χρήση distinct για απάλειψη διπλοτύπων
select pid, name
from performer, track
where performer.pid = track.pid and pos > 11;
```

Query 11

```
-- όλοι οι τίτλοι τραγουδιών που βρίσκονται
-- στη 10η θέση των cd
select stitle, pid, cid
from track, song
where track.sid = song.sid and pos = 10;
```

Query 12

```
-- φυσική σύζευξη όλων των πινάκων και
-- ταξινόμηση βάσει τίτλου τραγουδιού σε
-- φθίνουσα σειρά - μπορούμε και πολλαπλή
-- ταξινόμηση
select name, ctitle, stitle, year
from performer, song, track, cd
where cd.cid = track.cid and performer.pid =
        track.pid and song.sid = track.sid
order by stitle desc;
```

Query 13

```
-- τεχνητά πεδία στην απάντηση με χρήση
-- συναρτήσεων και πράξεων
select ctitle, ceil(year/100)
from cd;
```

Query 14

```
-- ίδιο με το Query 12 αλλά με χρήση μεταβλητών
-- ως εναλλακτικών ονομάτων πινάκων
select name, ctitle, stitle, year
from performer p, song s, track t, cd c
where c.cid = t.cid and p.pid = t.pid and
        s.sid = t.sid;
```

Query 15

```
-- σύζευξη πίνακα με τον εαυτό του - αλλά
-- παίρνω άχρηστη και περιττή πληροφορία
-- πρώτα προσθέτω c1.cid <> c2.cid και μετά
-- αλλάζω τη συνθήκη σε < ή >
select c1.ctitle, c1.year, c2.ctitle, c2.year
from cd c1, cd c2
where c1.year = c2.year;
```

Query 16

```
-- union και union all
select ctitle from cd where ctitle like 'S%'
union
select stitle from song where stitle like 'S%'
order by ctitle;
```

Query 17

```
-- τα cd που περιέχουν κάποιο ομώνυμο
-- τραγούδι, intersect
-- ΠΡΟΣΟΧΗ: η MySQL δεν υποστηρίζει το
-- intersect
select ctitle from cd
intersect
select stitle from song;
```

Query 18

```
-- εναλλακτική υλοποίηση του intersect
select ctitle
from song, cd
where stitle = ctitle;
```

Query 19

```
-- τα cd που ΔΕΝ περιέχουν κάποιο ομώνυμο
-- τραγούδι, difference
-- ΠΡΟΣΟΧΗ: η MySQL δεν υποστηρίζει το
-- except και η Oracle χρησιμοποιεί το minus
select ctitle from cd
except
select stitle from song;
```

Query 20

```
-- εναλλακτική υλοποίηση του except
-- (δεν λύνεται με σύζευξη)
select distinct cd.ctitle
from song, cd
where stitle <> ctitle;
```

Query 21

```
-- εμφωλευμένο αίτημα με in, εναλλακτικός
-- τρόπος για intersect
select ctitle
from cd
where ctitle in (select stitle from song);
```

Query 22

```
-- εμφωλευμένο αίτημα, εναλλακτικός τρόπος
-- για difference
select ctitle
from cd
where ctitle not in (select stitle from song);
```

Query 23

```
-- τα cd που δεν είναι μοναδικά σε κάποια
-- χρονιά - εμφωλευμένο αίτημα με exists,
-- προσθήκη c1.cid <> c2.cid
select *
from cd c1
where exists
        (select *
        from cd c2
        where c1.year = c2.year);
```

Query 24

```
-- εύρεση μεγαλύτερης (ή μικρότερης) χρονιάς
select year
from cd c1
where not exists
        (select *
        from cd c2
        where c2.year > c1.year);
```

Query 25

```
-- πρώτα διπλότυπα,
-- δεν υπάρχει λύση με σύζευξη
select c1.ctitle, c1.year
from cd c1, cd c2
where c1.year > c2.year;
```

Query 26

```
-- χρήση all, εύρεση max year
-- ΠΡΟΣΟΧΗ: δεν δουλεύει στην Oracle εξαιτίας
-- του NULL
select ctitle, year
from cd
where year >= all (select year from cd);
```

Query 27

```
-- δουλεύει μόνο αν το max (ή min) year είναι
-- μοναδικό και δεν έχουμε τιμές NULL!!!
select ctitle, year
from cd c1
where year < all
        (select year
        from cd c2
        where c2.cid <> c1.cid);
```

Query 28

```
-- χρήση any, (< all) == (! >= any)
select ctitle, year
from cd c1
where not year >= any
        (select year
        from cd c2
        where c2.year is not null and c2.cid <>
c1.cid);
```

Query 29

```
-- οι ερμηνευτές που το όνομά τους είναι
-- ταξινομικά μεγαλύτερο από όλους τους τίτλους
-- cd που έχουν κυκλοφορήσει
select *
from performer p1
where name < all
        (select ctitle
        from cd c, track t
        where c.cid=t.cid and t.pid=p1.pid);
```

Query 30

```
-- ίδιο με το Query 29: μπορούμε και χωρίς το any
-- και το all - χρησιμοποιούμε exists
select *
from performer p1
where not exists
        (select *
        from cd c, track t
        where c.cid=t.cid and t.pid=p1.pid
                and p1.name > c.ctitle);
```

Query 31

```
-- υλοποίηση του Query 22 (difference) με any
-- αντί του exists - λάθος η συγκεκριμένη
-- υλοποίηση, χρειάζεται "not ctitle = any"
select ctitle
from cd
where ctitle <> any (select stitle from song);
```

Query 32

```
-- τα cd του 21 αιώνα
select ctitle, ceil(year/100)
from cd
where ceil(year/100) > 20;
```

Query 33

```
-- όπως το Query 32 με select στο from!
select ctitle, century
from
        (select ctitle, ceil(year/100) as century
        from cd) cdnew
where century > 20;
```

Query 34

```
-- για κάθε cd το τραγούδι με τον ταξινομικά
-- μικρότερο τίτλο
select ctitle, stitle
from cd c, track t, song s
where c.cid = t.cid and t.sid = s.sid
        and s.stitle <= all
                (select stitle
                from track t, song s
                where c.cid = t.cid and t.sid = s.sid);
```

Query 35

```
-- όπως το Query 34 με χρήση select στην select!
-- (προσοχή στο distinct)
select ctitle,
        (select distinct stitle from track t, song s
        where cd.cid = t.cid and t.sid = s.sid
        and s.stitle <= all
                (select stitle from track t, song s
                where cd.cid = t.cid and
                        t.sid = s.sid)) as orderedfirst
from cd;
```

Query 36

```
-- πρόβλημα όταν το υποερώτημα στην
-- select επιστρέφει πολλές τιμές
select ctitle,
        (select distinct stitle from track t, song s
        where cd.cid = t.cid and t.sid = s.sid) as
songs
from cd;
```

Τέλος Ενότητας