

**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ**



**ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

Πανεπιστήμιο Ιωαννίνων
ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

2^ο Εξάμηνο 2019 – 2020



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Select → εμφάνιση - επιλογή

From → απο

Where → συνθήκη

Tips:

Πίνακες: Κεφαλαία

Εντολές: Κεφαλαία

Υπόλοιπα: Μικρά

Δημιουργία πίνακα:
CREATE TABLE όνομα πίνακα

Για να δούμε το σχήμα (δηλαδή την περιγραφή) ενός πίνακα:
DESCRIBE <όνομα πίνακα>;

Για να προσθέσουμε ένα πεδίο (μία στήλη) σε έναν :
ALTER TABLE <όνομα πίνακα> **ADD** <όνομα νέου πεδίου> <τύπος νέου πεδίου>;

Για να αλλάξουμε τον τύπο ενός πεδίου:
ALTER TABLE <όνομα πίνακα> **MODIFY** <όνομα πεδίου> <τύπος νέου πεδίου>;

Για να διαγράψουμε ένα πεδίο από έναν πίνακα, διαγράφεται ολόκληρη η στήλη άρα το γνώρισμα διαγράφεται:

ALTER TABLE <όνομα πίνακα> **DROP COLUMN** <όνομα πεδίου>;

Αν θέλουμε να διαγράψουμε από τη βάση μας έναν ολόκληρο πίνακα, ομοίως με το προηγούμενο:

DROP TABLE <όνομα πίνακα>;

Για να εισάγουμε δεδομένα σε πίνακες:
INSERT INTO <όνομα πίνακα> **VALUES** (<λίστα τιμών>);

Για να δούμε τα περιεχόμενα ενός πίνακα:
SELECT <λίστα πεδίων> **FROM** <όνομα πίνακα>;

ή
SELECT <λίστα πεδίων> **FROM** <όνομα πίνακα> **WHERE** <συνθήκη>;

Για την τροποποίηση των δεδομένων των πινάκων μας υπάρχουν οι εντολές
UPDATE, η οποία τροποποιεί πλειάδες που ήδη υπάρχουν σε έναν πίνακα

UPDATE <όνομα πίνακα>
SET <όνομα πεδίου> = <νέα τιμή> **WHERE** <συνθήκη>

Διαγράφει το περιεχόμενο των πλειάδων δηλαδή το γνώρισμα υπάρχει αλλά
είναι κενό:

DELETE FROM <όνομα πίνακα> **WHERE** <συνθήκη>;

Αποθήκευση αλλαγών:

COMMIT;

Για να δημιουργήσουμε ξένα κλειδιά, προσθέτουμε στην εντολή **CREATE TABLE** γραμμές της μορφής:

foreign key (<γνώρισμα ξένο-κλειδί>) **references** <όνομα πίνακα που αναφερόμαστε>(<γνώρισμα στο οποίο αναφερόμαστε>)

Για να δούμε συγκεκριμένα δεδομένα που είναι αποθηκευμένα στη βάση μας, υποβάλλουμε στη βάση μας ένα ερώτημα (query):

SELECT <λίστα πεδίων> **FROM** <λίστα πινάκων> **WHERE** <συνθήκη>;

Στο **WHERE** μπορεί να έχουμε σύνθετες συνθήκες, δηλαδή πολλές συνθήκες που συνδέονται με λογικούς τελεστές:

AND: Το λογικό «και» (ή αλλιώς «τομή»)

OR: Το λογικό «ή» (ή αλλιώς «ένωση»)

NOT: Η λογική άρνηση (ή αλλιώς «συμπλήρωμα»)

Υπάρχουν περιπτώσεις στις οποίες χρειάζεται να βρούμε πεδία που έχουν παρόμοιες τιμές και όχι ακριβώς ίσες:

LIKE

Τι χρησιμοποιούμε για το "οτιδήποτε":

%

Μεταφέρει την αλλαγή από ένα στοιχείο ενός πίνακα, στο ίδιο στοιχείο που υπάρχει και σε άλλους πίνακες:

CONSTRAINT

Μία ειδική μεταβλητή της βάσης μας που είναι τύπου date και περιέχει την τρέχουσα ημερομηνία:

SYSDATE

Ταξινόμηση αποτελέσματος:

```
SELECT <λίστα πεδίων>  
FROM <λίστα πινάκων>  
WHERE <συνθήκη>  
ORDER BY <λίστα πεδίων>;
```

Αύξουσα ταξινόμηση:

ASC

Φθίνουσα ταξινόμηση:

DESC

Για να δείξουμε κάποιο σύνολο:

IN

Δίνει τις τιμές που βρίσκονται ανάμεσα - μεταξύ από:

BETWEEN

Χρησιμοποιείται για να συγκρίνουμε τις τιμές **ενός** πεδίου με τις τιμές του αποτελέσματος που προκύπτει από μία εμφωλευμένη SELECT υπο-ερώτηση:
SOME

Χρησιμοποιείται για να συγκρίνουμε τις τιμές ενός πεδίου με **όλες** τις τιμές του αποτελέσματος που προκύπτει από μία εμφωλευμένη SELECT υπο-ερώτηση:
ALL

Η SQL προσφέρει ένα σύνολο συναρτήσεων που μπορούν να εφαρμοστούν σε ένα σύνολο πλειάδων και να υπολογίσουν μία αριθμητική τιμή από αυτό.

Τέτοιες συναρτήσεις είναι:

- Το ελάχιστο (**min**)
- Το μέγιστο (**max**)
- Το άθροισμα (**sum**)
- Ο μέσος όρος (**avg**)
- Το πλήθος (**count**)

Για να χωρίσουμε τις πλειάδες μας σε μικρότερες ομάδες:

```
SELECT <λίστα πεδίων>  
FROM <λίστα πινάκων>  
WHERE <συνθήκη>  
GROUP BY <λίστα πεδίων>;
```

Για να διαλέξουμε μόνο κάποιες από τις ομάδες:

```
SELECT <λίστα πεδίων>  
FROM <λίστα πινάκων>  
WHERE <συνθήκη>  
GROUP BY <λίστα πεδίων>  
HAVING <συνθήκη>;
```

Tips:

όταν η εκφώνηση λέει "βρείτε το ..." είναι **SELECT**
όταν η εκφώνηση λέει "για κάθε ..." είναι **GROUP BY**

Παραδείγματα με χρήση των εντολών

Παράδειγμα 1:

Δημιουργήστε ένα πίνακα με όνομα **ACTORS** και καταχωρήστε το **id**, **name** (όχι κενό), **birthdate**, **Oscars** και βάλτε και πρωτεύον κλειδί:

```
CREATE TABLE ACTORS (  
    ID varchar(10),  
    NAME varchar(20) not null,  
    BIRTHDATE date,  
    BIRTHPLACE varchar(20),  
    OSCARS int,  
    primary key (ID)  
);
```

Παράδειγμα 2:

Εμφάνισε το name, surname όταν το υπάρχει το A.M. και στους δύο πίνακες και το Final > 7

Students

A.M.	Name	Surname
....
....

Dbi

A.M.	Lab	Ex.	Final
....
....

```
Select Name, Surname  
From Students, Dbi  
Where 'Students'.A.M.=Dbi.Am AND Final>7
```

Παραδείγματα με εντολές:

Εμφάνιση σχήματος πίνακα ACTORS:

DESCRIBE ACTORS;

Προσθέστε στο πίνακα MOVIES το DURATION:

ALTER TABLE MOVIES **ADD** DURATION varchar(6);

Αλλάξτε το τύπο του DURATION σε int:

ALTER TABLE MOVIES **MODIFY** DURATION int;

Διαγράψτε το DURATION από το πίνακα:

ALTER TABLE MOVIES **DROP COLUMN** DURATION;

Διαγράψτε το πίνακα ACTOR:

DROP TABLE ACTOR

Εισάγετε στο πίνακα ACTORS μερικές πληροφορίες:

INSERT INTO ACTORS **VALUES** ('A01', 'Brad Pitt', '1963-12-18', 'Oklahoma', NULL);

ή

INSERT INTO ACTORS (ID, NAME) **VALUES** ('A02', 'George Clooney');

Παραδείγματα με χρήση εντολών:

Να εμφανίσετε όλο το πίνακα **ACTORS**:

SELECT * FROM ACTORS;, * → όλος ο πίνακας

Να επιλέξετε από το πίνακα **ACTORS**, το **name**, **oscars**:

SELECT NAME, OSCARS FROM ACTORS;

Αλλάξτε από το πίνακα **ACTORS** το **birthdate** με **id = 'A02'** σε **1968-08-15**:

UPDATE ACTORS SET BIRTHDATE = '1968-08-15' WHERE ID = 'A02';

Διαγράψτε από το πίνακα **MOVIES** τα περιεχόμενα με **year <= 1960**:

DELETE MOVIES WHERE YEAR <= 1960;

Διαγράψτε το πίνακα **MOVIES**:

DELETE MOVIES;

Βάλτε ξένο κλειδί το **id** από το πίνακα **ACTORS**:

foreign key (ID) references ACTORS(ID)

Παραδείγματα με χρήση εντολών:

Επιλέξτε το id, name, birthdate, από το πίνακα ACTORS όπου το birthdate > 1950-01-01:

```
SELECT ID, NAME, BIRTHDATE FROM ACTORS WHERE BIRTHDATE > '1950-01-01';
```

Τροποποίησε από το πίνακα ACTORS το Oscar όπου το name = '...' και surname = '...' και κάνε το = 10

```
UPDATE ACTORS  
SET oscars=10  
WHERE name = '...' AND surname = '...';
```

Τροποποίησε από το πίνακα ACTORS το birthdate όπου το id = 2 και κάνε το = 1968-08-15

```
UPDATE ACTORS  
SET birthdate = '1968-08-15'  
WHERE id = '2';
```

Τροποποίησε από το πίνακα MOVIES το tickets όπου το ticket = sow και πρόσθεσέ το 10000

```
UPDATE MOVIES  
SET tickets = tickets + 10000  
WHERE title = 'sow';
```

Παραδείγματα με χρήση εντολών:

Τροποποίησε από το πίνακα MOVIES το budget όπου το budget > 100000 και πολλαπλασίασε το με το 0.7 και το ανάποδο:

Για >100.000 --> *0.7

Για <100.000 --> *1.2

UPDATE MOVIES

SET budget = budget * 0.7

WHERE budget > 100.000;

UPDATE MOVIES

SET budget = budget * 1.2

WHERE budget < 100.000;

Παραδείγματα με χρήση εντολών:

Δημιουργείτε μία νέα πλειάδα στη βάση σας για την ηθοποιό Nicole Kidman που γεννήθηκε στις 20 Ιουνίου 1967. Ο αριθμός ταυτότητάς της είναι A03. Στη συνέχεια, τροποποιήστε την πλειάδα ώστε να καταχωρηθεί στη βάση η πληροφορία ότι η συγκεκριμένη ηθοποιός έχει κερδίσει 1 βραβείο oscar.

```
UPDATE MOVIES  
SET budget = budget * 1.1;
```

```
{  
INSERT INTO ACTORS (id,name,surname,birthdate,birthplace,oscars) VALUES  
('100','nicole','kidman','1967-6-20','newwork',NULL);
```

```
UPDATE ACTORS  
SET oscars = '1'  
WHERE id = '100';  
}
```

Αυξήστε κατά 20% τα κόστη όλων των ταινιών με σκηνοθέτη τον Steven Spielberg.

```
UPDATE MOVIES  
SET budget = budget * 1.2  
WHERE director = ...';
```

Παραδείγματα με χρήση εντολών:

Επιλέξτε το title, budget από το πίνακα MOVIES όπου ο director να έχει μέσα και το Spielberg:

```
SELECT TITLE, BUDGET  
FROM MOVIES  
WHERE DIRECTOR LIKE '%Spielberg%';
```

Παραδείγματα με χρήση εντολών:

Αλλάξτε το id σε int (30) και δημιουργήστε ένα πίνακα STARS με μερικά στοιχεία και προσθέστε μερικά πράγματα στο STARS στην συνέχεια εισάγετε μερικά δεδομένα:

ALTER TABLE ACTORS MODIFY id int (30); → αλλαγή του id σε int(30)

CREATE TABLE STARS (→ δημιουργία πίνακα STARS
id varchar (30),
title varchar (30)
role varchar (.....);

ALTER TABLE STARS ADD CONSTRAINT fkey foreign key (id) references ACTORS (id);

ALTER TABLE STARS ADD CONSTRAINT fkey2 foreign key (title) references MOVIES (title);

INSERT INTO STARS (id,title,role,cache) **VALUES** ('1','sow','protag','1'); → εισαγωγή δεδομένων

Παραδείγματα Χρήσης Εντολών με AND, OR, LIKE, SYSDATE:

Επέλεξε το title από το MOVIES όπου το year = 2009 ή year = 2011:

```
SELECT title  
FROM MOVIES  
WHERE year = 2009 OR year = 2011;
```

Επέλεξε το title, ticket από το MOVIES όπου το year = 2009 ή year = 2011 και budget > 10000000:

```
SELECT TITLE, TICKETS  
FROM MOVIES  
WHERE (year = 2009 OR year = 2011) AND budget > 10000000;
```

Επέλεξε το title, ticket από το MOVIES όπου ο director να έχει μέσα και το ast:

```
SELECT TITLE, BUDGET  
FROM MOVIES  
WHERE director LIKE '%ast%';
```

Παραδείγματα Χρήσης Εντολών LIKE, Ταξινόμηση:

Επιλέξτε το title από το MOVIES όπου το year = 2011 και ο title να έχει μέσα το α:

```
SELECT title  
FROM MOVIES  
WHERE year = 2011 AND title LIKE 'α%';
```

Επέλεξε όλα τα στοιχεία από το πίνακα ACTORS όπου το birthdate < sysdate:

```
SELECT *  
FROM ACTORS  
WHERE BIRTHDATE < 'SYSDATE';
```

Επέλεξε id 'Ar. Tautotitas', name 'Onoma', birthplace 'Katagwgi' από το ACTORS όπου το birthplace έχει μέσα και το USA

```
SELECT id 'Ar. Tautotitas', name 'Onoma', birthplace 'Katagwgi'  
FROM ACTORS  
WHERE birthplace LIKE '%USA%';
```

Παραδείγματα Χρήσης Εντολών AND, BETWEEN:

Επέλεξε όλα τα στοιχεία από το πίνακα ACTORS και κάνε αύξουσα και φθίνουσα ταξινόμηση με βάση το name:

```
SELECT *  
FROM ACTORS  
ORDER BY name ASC;
```

```
SELECT *  
FROM ACTORS  
ORDER BY NAME DESC;
```

Εμφανίστε τα στοιχεία για τους ηθοποιούς που το όνομά τους είναι (ακριβώς) είτε «Brad» είτε «Angelina»:

```
SELECT *  
FROM ACTORS  
WHERE NAME IN ('Brad', 'Angelina');
```


Παραδείγματα Χρήσης Εντολών:

Εμφανίστε τα στοιχεία για το MOVIES όπου το year είναι μεταξύ του 1999 και 2000:

```
SELECT *  
FROM MOVIES  
WHERE year BETWEEN 1999 AND 2000;
```

Εμφάνισε το title από το MOVIES όπου το year είναι 2010 ή 2012 ή 2014 ή 2016 και director είναι ast ή asdffsd και budget < 100000:

```
SELECT title  
FROM MOVIES  
WHERE (year = 2010 or year = 2012 or year = 2014 or year = 2016) AND (director = 'ast' or  
director = 'asdffsd') AND (budget < 100000);  
Η
```

```
SELECT title  
FROM MOVIES  
WHERE year IN (2010, 2012, 2014, 2016) AND director LIKE 'ast' or director LIKE 'asdffsd')  
AND (budget < 100000);
```

Παραδείγματα Χρήσης Εντολών:

Θέλουμε να εμφανίσει τους αριθμούς ταυτότητας και τα ονόματα των ηθοποιών που έχουν πρωταγωνιστήσει στην ταινία sow:

```
SELECT ID, NAME  
FROM ACTORS  
WHERE ID IN (SELECT ID  
FROM STARS  
WHERE TITLE = 'sow');
```

Θέλουμε να εμφανίσουμε τους τίτλους και το έτος παραγωγής των ταινιών οι οποίες κόστισαν περισσότερο από κάποια (οποιαδήποτε) ταινία που έχει σκηνοθετήσει τον ast:

```
SELECT TITLE, YEAR  
FROM MOVIES  
WHERE BUDGET > SOME (SELECT BUDGET  
FROM MOVIES  
WHERE DIRECTOR= 'ast');
```

Εμφανίστε τους αριθμούς ταυτότητας των ηθοποιών που έχουν συμπρωταγωνιστήσει σε μία τουλάχιστον ταινία με τον ηθοποιό που έχει αριθμό ταυτότητας 1:

```
SELECT id  
FROM STARS  
WHERE title IN (SELECT title  
FROM STARS  
WHERE id = '1');
```

Παραδείγματα Χρήσης Εντολών:

Θέλουμε να εμφανίσουμε τους τίτλους και το έτος παραγωγής των ταινιών οι οποίες κόστισαν περισσότερο από κάθε ταινία (δηλαδή όλες) που κυκλοφόρησε το 2011:

```
SELECT TITLE, YEAR  
FROM MOVIES  
WHERE BUDGET > ALL (SELECT BUDGET  
FROM MOVIES  
WHERE YEAR = 2011);
```

Εμφανίστε τους αριθμούς ταυτότητας των ηθοποιών που δεν έχουν πρωταγωνιστήσει σε καμία ταινία με σκηνοθέτη τον ast:

```
SELECT id  
FROM STARS  
WHERE title NOT IN (SELECT title  
FROM MOVIES  
WHERE director = 'ast');
```

Παράδειγμα χρήσης εντολών:

Θέλουμε να βρούμε όλους τους αριθμούς ταυτότητας των ηθοποιών που έχουν κερδίσει περισσότερα από 2 oscar ή έχουν αμειφθεί με περισσότερα από €10.000.000 για κάποιον ρόλο τους:

```
(SELECT ID
FROM ACTORS
WHERE OSCARS > 2)
UNION
(SELECT ID
FROM STARS
WHERE CACHE > 10000000);
```

Θέλουμε να βρούμε όλους τους αριθμούς ταυτότητας των ηθοποιών που έχουν κερδίσει περισσότερα από 2 oscar και έχουν πρωταγωνιστήσει στην ταινία sow:

Αντί της **INTERSECT**:

```
SELECT ID
FROM ACTORS
WHERE OSCARS > 2 AND ID IN(
SELECT ID
FROM STARS
WHERE title = 'sow');
```

Παράδειγμα χρήσης εντολών:

Θέλουμε να βρούμε όλους τους αριθμούς ταυτότητας των ηθοποιών που έχουν κερδίσει περισσότερα από 2 oscar αλλά δεν έχουν πρωταγωνιστήσει στην ταινία sow:

Αντί της **MINUS**:

```
SELECT ID
FROM ACTORS
WHERE OSCARS > 2 AND ID NOT IN(
  SELECT ID
  FROM STARS
  WHERE title = 'sow');
```

Παράδειγμα χρήσης εντολών: Ερωτήσεις SELECT με πολλούς πίνακες:

Πρέπει να συνδέει μεταξύ τους πλειάδες των δύο πινάκων που αναφέρονται στον ίδιο ηθοποιό:

```
SELECT NAME, TITLE  
FROM ACTORS, STARS  
WHERE ACTORS.ID = STARS.ID;
```

Αν θέλουμε επιπλέον να εμφανίσουμε το όνομα του σκηνοθέτη κάθε ταινίας στην οποία πρωταγωνιστεί κάθε ηθοποιός, θα χρειαστούμε και τον πίνακα MOVIES:

```
SELECT ACTORS.NAME, STARS.TITLE, MOVIES.DIRECTOR  
FROM ACTORS, STARS, MOVIES  
WHERE ACTORS.ID = STARS.ID AND MOVIES.TITLE = STARS.TITLE;
```

Παράδειγμα χρήσης εντολών:

Εμφανίστε τους αριθμούς ταυτότητας και τα ονόματα των ηθοποιών που έχουν πρωταγωνιστήσει σε ταινία με σκηνοθέτη τον ast:

```
SELECT ACTORS.id, ACTORS.name  
FROM ACTORS, STARS, MOVIES  
WHERE ACTORS.id = STARS.id AND MOVIES.director = 'ast';
```

Παράδειγμα χρήσης εντολών:

Βρείτε το μέσο όρο κόστους των ταινιών:

```
SELECT avg(BUDGET)
FROM MOVIES;
```

Βρείτε το πλήθος των ηθοποιών:

```
SELECT count(*)
FROM ACTORS;
```

```
SELECT count(ID)
FROM STARS;
```

Διακριτό πλήθος:

```
SELECT count(DISTINCT ID)
FROM STARS;
```


Παράδειγμα χρήσης εντολών:

Βρείτε την μεγαλύτερη και την μικρότερη αμοιβή που έχει πάρει ποτέ ο ηθοποιός με αριθμό ταυτότητας A08 για ρόλο που έχει παίξει.

```
SELECT max(cache), min(cache)
FROM STARS
WHERE ID = 'A08';
```

```
SELECT title, avg(CACHE)
FROM STARS
GROUP BY title;
```

πρώτα χωρίζει τις πλειάδες του πίνακα STARS σε ομάδες ανάλογα με την τιμή τους στο πεδίο TITLE (μία ομάδα για όλους τους πρωταγωνιστικούς ρόλους με τιμή στο TITLE ίση με Titanic, μία για αυτούς με τιμή ίση με Armagedon κτλ.) και μετά εφαρμόζει την συναθροιστική συνάρτηση σε κάθε ομάδα χωριστά. Αν δηλαδή υπάρχουν 5 διαφορετικές ταινίες στον πίνακα STARS, τότε η παραπάνω ερώτηση θα δώσει 5 διαφορετικούς μέσους όρους, έναν για κάθε ομάδα. Για να ξεχωρίζουμε ποια τιμή αντιστοιχεί σε κάθε ομάδα, χρειάζεται στο SELECT να βάζουμε και το πεδίο που υπάρχει στο GROUP BY. Προσοχή: το μόνο πεδίο που μπορεί να υπάρχει στη SELECT είναι το πεδίο με βάση το οποίο γίνεται η ομαδοποίηση, και φυσικά και όποιες συναρτήσεις θέλουμε να υπολογισθούν.

Παράδειγμα χρήσης εντολών:

Εμφάνιση του πλήθους των ταινιών με λιγότερα από 1.000.000 εισιτήρια ανά σκηνοθέτη:

```
SELECT DIRECTOR, COUNT(*)  
FROM MOVIES  
WHERE TICKETS < 1000000  
GROUP BY DIRECTOR;
```

Εμφάνιση του πλήθους των πρωταγωνιστών ανά σκηνοθέτη και ανά ταινία:

```
SELECT DIRECTOR, STARS.TITLE, COUNT(*)  
FROM STARS, MOVIES  
WHERE MOVIES.TITLE = STARS.TITLE  
GROUP BY DIRECTOR, STARS.TITLE;
```

Βρείτε για κάθε ταινία τον συνολικό αριθμό των oscars που έχουν κερδίσει οι πρωταγωνιστές της:

```
SELECT title, sum(oscars)  
FROM ACTORS, STARS  
WHERE ACTORS.ID = STARS.ID  
GROUP BY title;
```

Παράδειγμα χρήσης εντολών:

Θέλουμε να βρούμε τους αριθμούς ταυτότητας και τον αριθμό των ταινιών στις οποίες έχει πρωταγωνιστήσει κάθε ηθοποιός, μόνο για τους ηθοποιούς που έχουν πρωταγωνιστήσει σε περισσότερες από 15 ταινίες:

```
SELECT ID, COUNT(*)  
FROM STARS  
GROUP BY ID  
HAVING COUNT(*) > 15;
```

Για να εμφανίσουμε τον αριθμό ταινιών που έχει σκηνοθετήσει κάθε σκηνοθέτης, μόνο για τους σκηνοθέτες οι οποίοι έχουν σκηνοθετήσει περισσότερες ταινίες από τον σκηνοθέτη ast:

```
SELECT DIRECTOR, COUNT(*)  
FROM MOVIES  
GROUP BY DIRECTOR  
HAVING COUNT(*) > (SELECT COUNT(*)  
FROM MOVIES  
WHERE DIRECTOR = 'ast');
```

EXTRA Ασκήσεις:

- Βρείτε τα καταστήματα που αγοράζουν από τον προμηθευτή με όνομα A:

```
SELECT shop.id, shop.name  
FROM SHOP, BUYING, SUPPLIER  
WHERE shop.id = buying.id AND buying.id = supplier.sid  
AND supplier.name = 'A'
```

SHOP (id ,name ,address ,manager)

PRODUCT (pname ,company ,pcity)

SUPPLIER (sid ,sname ,scity)

BUYING (id ,sid) (καταστήματα αγοράζουν από προμηθευτές)

SUPPLYING (sid ,pname) (έμποροι προμηθεύουν προϊόντα)

- Βρείτε τα καταστήματα τα οποία είτε έχουν manager τον 'B' είτε αγοράζουν από προμηθευτές που έχουν αυτό το όνομα

```
SELECT shop.id
FROM SHOP
WHERE manager = 'B'
UNION
SELECT shop.id
FROM shp, supplier, buying
WHERE shop.id = buying.id AND supplier.sid = buying.cid AND supplier.name = 'B'
```

ή

```
SELECT shop.id
FROM shp, supplier, buying
WHERE shop.id = buying.id AND
supplier.sid = buying.cid AND supplier.name
= 'B' OR manager = 'B'
```

SHOP (id ,name ,address ,manager)

PRODUCT (pname ,company ,pcity)

SUPPLIER (sid ,sname ,scity)

BUYING (id ,sid) (καταστήματα αγοράζουν από προμηθευτές)

SUPPLYING (sid ,pname) (έμποροι προμηθεύουν προϊόντα)

- Βρείτε τους αριθμούς των παραγγελιών για τα προϊόντα που παραγγέλλονται σε ποσότητα μικρότερη του 100 και από πελάτες που βρίσκονται στην Αθήνα

```
SELECT orderno
FROM client, order
WHERE client.cid = order.cid AND quantity < 100 AND city = "Athens"
```

ή

```
SELECT orderno
FROM order
WHERE quantity < 100 AND cid IN (SELECT cid
FROM clients
WHERE city = "Athens")
```

PRODUCT (pid ,stock , supplier)

CLIENT (cid ,name ,address ,city)

ORDER(orderno , date ,quantity, pid, cid)

- Βρείτε τα ονόματα και τις διευθύνσεις των πελατών οι οποίοι δίνουν παραγγελία για προϊόντα για τα οποία δεν υπάρχει stock

```
SELECT name, address, client.cid  
FROM client, order, product  
WHERE product.pid = orders.pid AND orders.cid =  
client.cid AND client.stock = 0
```

ή

```
SELECT name, address, cid  
FROM clients  
WHERE cid IN (SELECT cid FROM order WHERE pid  
IN (SELECT pid FROM product WHERE stock = 0))
```

PRODUCT (pid ,stock , supplier)

CLIENT (cid ,name ,address ,city)

ORDER(orderno , date ,quantity, pid, cid)