

Σημειώσεις στο μάθημα Βάσεις Δεδομένων

Hands on 01

Πίνακας περιεχομένων

1	Γενικά	3
1.1	Εισαγωγή.....	3
1.2	SQL	5
1.3	Το Human Resources (HR) Schema.....	6
2	Εντολές SQL	7
2.1	Ανάκτηση Δεδομένων με την χρήση του SELECT	7
2.2	Χρήση αριθμητικών τελεστών στο SELECT	8
2.3	Χρήση του NULL στο SELECT	9
2.4	Ψευδώνυμα στηλών στο SELECT	9
2.5	Ο χαρακτήρας συνένωσης, και ο τελεστής DISTINCT	10
2.6	Ασκήσεις στο SELECT	11

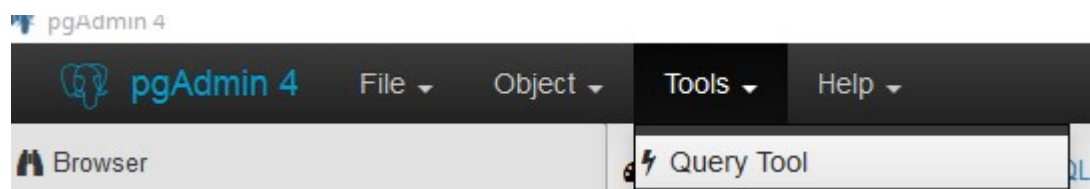
1 Γενικά

1.1 Εισαγωγή

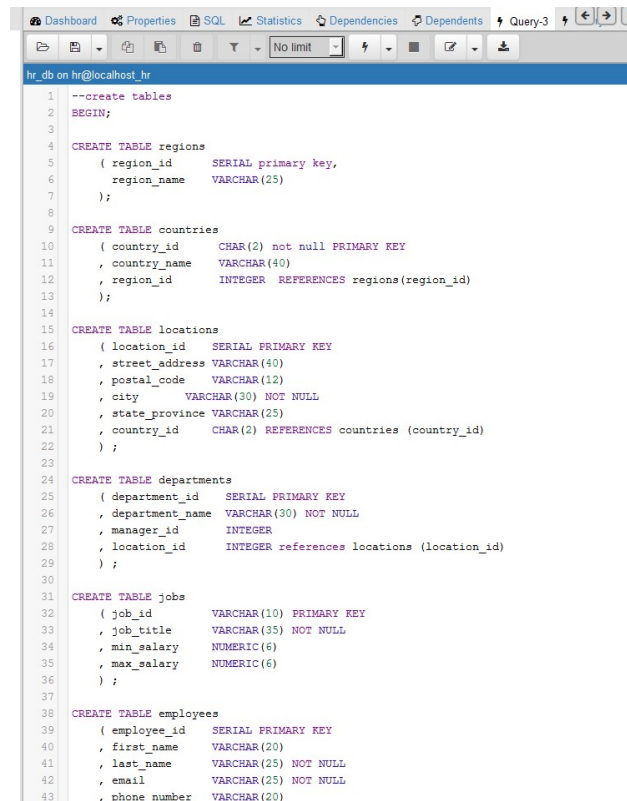
Σε αυτή την πρακτική εκπαίδευση θα μελετήσουμε πρακτικά βασικά θέματα των σχεσιακών βάσεων δεδομένων αλλά και θα δούμε διάφορες εντολές της SQL γλώσσας. Σκοπός είναι μετά την ολοκλήρωση της πρακτικής να μπορείτε να γράφετε ερωτήματα σε έναν ή περισσότερους πίνακες, να μπορείτε να τροποποιείτε δεδομένα και να δημιουργείτε αντικείμενα.

Στην περίπτωση

α. που έχετε επιλέξει το περιβάλλον του pgAdmin και αφού έχετε ολοκληρώσει τα προηγούμενα βήματα (postgresql_installation.01.oct2016, pgAdmin.01.oct2016) , συνδέεστε στον server σας χρήστης **hr** επιλέγετε την **HRProdDB** και επιλέγετε **Tools>Query Tool** οπότε ανοίγει ένα μια καρτέλα για την εκτέλεση εντολών σε SQL.




Στη συνέχεια ανοίγετε το αρχείο HR_pgsql.sql (με notepad) από το οποίο επιλέγετε ΟΛΟ το κείμενο και κάνετε αντιγραφή και επικόλληση μέσα στην καρτέλα SQL του pgAdmin που ανοίξατε προηγουμένως.



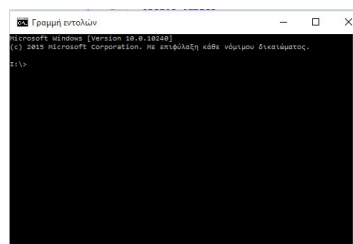
```

1  --create tables
2  BEGIN;
3
4  CREATE TABLE regions
5  (
6    region_id      SERIAL primary key,
7    region_name    VARCHAR(25)
8  );
9
10 CREATE TABLE countries
11 (
12   country_id      CHAR(2) not null PRIMARY KEY
13   , country_name  VARCHAR(40)
14   , region_id     INTEGER REFERENCES regions(region_id)
15 );
16
17 CREATE TABLE locations
18 (
19   location_id     SERIAL PRIMARY KEY
20   , street_address VARCHAR(40)
21   , postal_code    VARCHAR(12)
22   , city           VARCHAR(30) NOT NULL
23   , state_province VARCHAR(25)
24   , country_id     CHAR(2) REFERENCES countries (country_id)
25 );
26
27 CREATE TABLE departments
28 (
29   department_id   SERIAL PRIMARY KEY
30   , department_name VARCHAR(30) NOT NULL
31   , manager_id    INTEGER
32   , location_id    INTEGER references locations (location_id)
33 );
34
35 CREATE TABLE jobs
36 (
37   job_id          VARCHAR(10) PRIMARY KEY
38   , job_title      VARCHAR(35) NOT NULL
39   , min_salary     NUMERIC(6)
40   , max_salary     NUMERIC(6)
41 );
42
43 CREATE TABLE employees
44 (
45   employee_id     SERIAL PRIMARY KEY
46   , first_name     VARCHAR(20)
47   , last_name      VARCHAR(25) NOT NULL
48   , email          VARCHAR(25) NOT NULL
49   , phone number   VARCHAR(20)

```

Πατάτε το εικονίδιο  και περιμένετε για να ολοκληρωθεί η εκτέλεση των εντολών. Μετά από αυτό είστε έτοιμοι να ξεκινήσετε.

β. που έχετε επιλέξει το περιβάλλον του SQLDeveloper και αφού έχετε ολοκληρώσει τα προηγούμενα βήματα (oracleexpress_installation.01.oct2016, sqldeveloper.01.oct2016) , ανοίγετε μια γραμμή εντολών των windows στην οποία δίνετε:



sqlplus / as sysdba

sql>alter user hr identified by hr_passwd account unlock;

```

Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\>sqlplus / as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on -{V #41 31 11:19:12 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> alter user hr identified by hr_passwd account unlock;

User altered.

SQL>
  
```

Μετά από αυτό είστε έτοιμοι να ξεκινήσετε αφού δημιουργήσετε μια νέα σύνδεση αυτή τη φορά για τον χρήστη hr (ΟΧΙ για τον hr2).

1.2 SQL

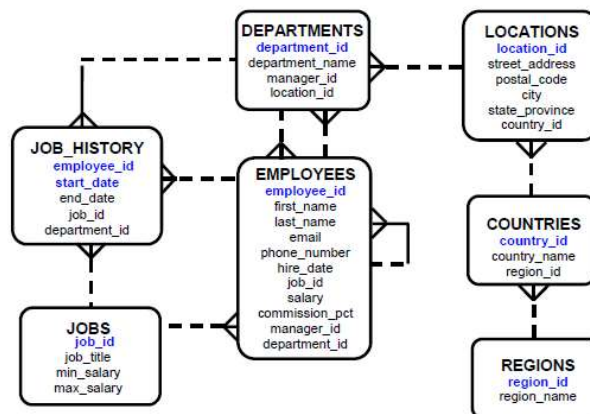
Όπως έχουμε ήδη αναφέρει με την χρήση της sql (structurequerylanguage) σε μια σχεσιακή βάση δεδομένων, αποκρύπτονται η διαδικαστικές λεπτομέρειες που απαιτούνται για την διαχείριση των δεδομένων. Η sql αποτελεί την επιλεγμένη από την ANSI (American National Standard Institute) και το ISO (International Standards Organization) γλώσσα για σχεσιακές βάσεις δεδομένων και προσφέρει την δυνατότητα για:

- εκτέλεση ερωτημάτων
- εισαγωγή, ενημέρωση και διαγραφή γραμμών από πίνακα
- δημιουργία, αντικατάσταση, μεταβολή και διαγραφή αντικειμένων
- έλεγχο πρόσβασης στην βάση δεδομένων και σε αντικείμενα
- διασφάλιση της ακεραιότητας και των περιορισμών

Μπορούμε να κατηγοριοποιήσουμε τις εντολές της SQL ανάλογα ως εξής:

Εντολή	Περιγραφή
SELECT INSERT UPDATE DELETE MERGE	Ανακτά, εισάγει, τροποποιεί και διαγράφει δεδομένα. Αναφέρεται ως <i>data manipulation language</i> (DML)
CREATE ALTER DROP RENAME TRUNCATE COMMENT	Ορίζει, μεταβάλει και διαγράφει δομές δεδομένων. Αναφέρεται ως <i>data definition language</i> (DDL)
GRANT REVOKE	Παραχωρεί ή αφαιρεί δικαιώματα πρόσβασης στην βάση δεδομένων και σε δομές της. Αναφέρεται ως <i>data control language</i> (DCL)
COMMIT ROLLBACK SAVEPOINT	Δίνει την δυνατότητα ελέγχου των αλλαγών που γίνονται από τις DML εντολές. Οι αλλαγές μπορούν να ομαδοποιηθούν σε λογικές συναλλαγές (logical transactions). Αναφέρεται ως <i>transaction control language</i> (TCL)

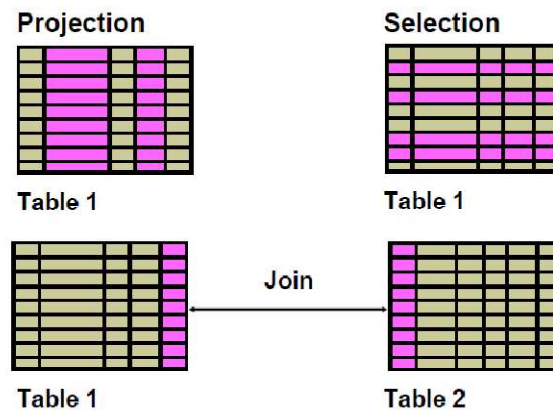
1.3 To Human Resources (HR) Schema



2 Εντολές SQL

2.1 Ανάκτηση Δεδομένων με την χρήση του SELECT

Για την ανάκτηση δεδομένων από την βάση δεδομένων χρησιμοποιείτε την εντολή SELECT της sql η οποία σας δίνει την δυνατότητα να περιορίσετε και τα πεδία τα οποία εμφανίζονται. Πιο συγκεκριμένα οι δυνατότητες που σας δίνονται με την χρήση του SELECT είναι



Προβολή (projection): με την οποία επιλέγετε να προβάλλονται μια ή περισσότερες στήλες (πεδία) της επιλογής σας.

Επιλογή (selection): με την οποία επιλέγετε της γραμμές του πίνακα που επιστρέφονται με το ερώτημα σας. Εδώ μπορούν να εφαρμοστούν διάφορα κριτήρια για τον περιορισμό των γραμμών.

Σύνδεση (joining): με την οποία ανακτούνται δεδομένα που βρίσκονται σε διαφορετικούς συσχετισμένους πίνακες. Μια βασική σύνταξη του SELECT δείχνεται στην παρακάτω εικόνα.

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM    table;
```

Μπορούμε να επιλέξουμε **όλες τις στήλες** από έναν πίνακα με την χρήση του συμβόλου *. Ένας άλλος τρόπος είναι να γράψουμε τα ονόματα όλων των στηλών του πίνακα αμέσως μετά το SELECT.

SELECT department_id, department_name, manager_id, location_id FROM departments;

```
SELECT *  
FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10 Administration	200	1700
2	20 Marketing	201	1800
3	30 Shipping	124	1500
4	40 IT	103	1400
5	50 Sales	149	2500
6	60 Executive	100	1700
7	70 Accounting	205	1700
8	80 Contracting	(null)	1700

Μπορούμε να επιλέξουμε **ορισμένες στήλες** από έναν πίνακα με την χρήση του συμβόλου γράφοντας τα ονόματα όλων των στηλών χωρισμένα με κόμμα του πίνακα που ενδιαφερόμαστε αμέσως μετά το SELECT.

```
SELECT location_id, department_id FROM departments;
```

Στη σύνταξη εντολών SQL, όπως είναι το SELECT, ισχύει ότι οι εντολές δεν είναι case sensitive, ότι μπορούμε να τις γράψουμε σε μια ή περισσότερες γραμμές, ότι οι λέξεις κλειδιά δεν μπορούν να διαχωριστούν μεταξύ γραμμών και ότι τερματίζονται με την χρήση του αγγλικού ερωτηματικού (;).

2.2 Χρήση αριθμητικών τελεστών στο SELECT

Κατά την σύνταξη του SELECT μπορούμε να πραγματοποιήσουμε και αριθμητικές πράξεις σε αριθμητικά δεδομένα ή σε ημερομηνίες με την χρήση τελεστών όπως + (πρόσθεση), - (αφαίρεση), * (πολλαπλασιασμός), / (διαίρεση). Η χρήση των τελεστών μπορεί να γίνει σε οποιοδήποτε σημείο εκτός από το FROM.

Παράδειγμα χρήσης αριθμητικών τελεστών είναι το παρακάτω όπου στις τιμές του πεδίου *salary* προσθέτουμε 300 (έστω ότι θέλουμε να δώσουμε αύξηση 300 ευρώ σε ΟΛΟΥΣ!!!).

```
SELECT last_name, salary, salary + 300  
FROM employees;
```

	LAST_NAME	SALARY	SALARY+300
1	King	24000	24300
2	Kochhar	17000	17300
3	De Haan	17000	17300
4	Hunold	9000	9300
5	Ernst	6000	6300
6	Lorentz	4200	4500
7	Mourgos	5800	6100
8	Rajs	3500	3800
9	Davies	3100	3400
10	Matos	2600	2900

...

Στην περίπτωση όπου μια αριθμητική παράσταση περιέχει περισσότερους του ενός τελεστές τότε ο πολλαπλασιασμός και η διαίρεση εκτελούνται πρώτοι. Εάν οι τελεστές έχουν την ίδια προτεραιότητα τότε η εκτέλεση γίνεται από αριστερά προς τα δεξιά. Τέλος με την χρήση παρενθέσεων μπορούμε να καθορίσουμε ποιες αριθμητικές παραστάσεις θα εκτελεστούν πρώτες. Για παράδειγμα το αποτέλεσμα των δύο παρακάτω εντολών sql διαφέρει (γιατί;).

SELECT last_name, salary, 12(salary+100) FROM departments;*

*SELECT last_name, salary, 12*salary+100 FROM departments;*

2.3 Χρήση του NULL στο SELECT

Το NULL δηλώνει ότι τιμή μιας στήλης σε μια γραμμή δεν μας είναι διαθέσιμη ή δεν την γνωρίζουμε ή δεν μπορεί να εφαρμοστεί. Διαφέρει από το τιμή μηδέν ή το κενό διάστημα.

SELECT last_name, job_id, salary, commission_pct FROM employees;				
	LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
1	King	AD_PRES	24000	(null)
2	Kochhar	AD_VP	17000	(null)
...				
12	Zlotkey	SA_MAN	10500	0.2
13	Abel	SA_REP	11000	0.3
14	Taylor	SA_REP	8600	0.2
...				
19	Higgins	AC_MGR	12000	(null)
20	Gietz	AC_ACCOUNT	8300	(null)

Η χρήση του null σε μια αριθμητική έκφραση μας επιστρέφει πάλι null π.χ.

SELECT last_name, 12*salary*commission_pct FROM employees;		
	LAST_NAME	12*SALARY*COMMISSION_PCT
1	King	(null)
2	Kochhar	(null)
...		
12	Zlotkey	25200
13	Abel	39600
14	Taylor	20640
...		
19	Higgins	(null)
20	Gietz	(null)

2.4 Ψευδώνυμα στηλών στο SELECT

Με τη χρήση των ψευδώνυμων σε στήλες μπορούμε να μετονομάσουμε μια στήλη (κατά την εκτέλεση του SELECT). Η λειτουργία αυτή είναι χρήσιμη κατά την εκτέλεση υπολογισμών. Πραγματοποιείτε είτε γράφοντας το ψευδώνυμο αμέσως μετά το όνομα της στήλης που θέλουμε να μετονομάσουμε ή γράφοντας την δεσμευμένη λέξη AS και μετά το ψευδώνυμο. Στην περίπτωση που αυτό περιέχει κενά ή ειδικούς χαρακτήρες είναι χρήση διπλών εισαγωγικών (π.χ. "SALARY INCR") οπότε και είναι case sensitive.

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

	NAME	COMM
1	King	(null)
2	Kochhar	(null)
3	De Haan	(null)

...

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

	Name	Annual Salary
1	King	288000
2	Kochhar	204000
3	De Haan	204000

...

2.5 Ο χαρακτήρας συνένωσης, και ο τελεστής DISTINCT

Ο χαρακτήρας συνένωσης δηλώνετε με το σύμβολο || (δύο κάθετες μπάρες) και χρησιμοποιείται για να συνενώσει τιμές στηλών ή χαρακτήρες σε άλλες στήλες (κατά την εκτέλεση του SELECT). Στην περίπτωση που γίνει συνένωση στήλης με NULL τότε το αποτέλεσμα είναι η τιμή της στήλης.

```
SELECT last_name || ' is a ' || job_id
AS "Employee Details"
FROM employees;
```

	Employee Details
1	Abel is a SA_REP
2	Davies is a ST_CLERK
3	De Haan is a AD_VP
4	Ernst is a IT_PROG
5	Fay is a MK_REP

...

18	Vargas is a ST_CLERK
19	Whalen is a AD_ASST
20	Zlotkey is a SA_MAN

Κατά την εκτέλεση του ερωτήματος η προκαθορισμένη λειτουργία είναι η εμφάνιση όλων των εγγραφών συμπεριλαμβανομένων και των διπλότυπων. Η χρήση του τελεστή DISTINCT μας δίνει την δυνατότητα να απαλείψουμε τις διπλότυπες εγγραφές. Τοποθετούμε τον τελεστή αυτόν αμέσως μετά την δεσμευμένη λέξη SELECT, πχ. SELECT DISTINCT ...

<pre>SELECT department_id FROM employees;</pre>	1						
<table> <tr> <th>DEPARTMENT_ID</th> </tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table>	DEPARTMENT_ID	1	2	3	4	5	
DEPARTMENT_ID							
1							
2							
3							
4							
5							
...							
<pre>SELECT DISTINCT department_id FROM employees;</pre>	2						
<table> <tr> <th>DEPARTMENT_ID</th> </tr> <tr><td>1 (null)</td></tr> <tr><td>2 90</td></tr> <tr><td>3 20</td></tr> <tr><td>4 110</td></tr> </table>	DEPARTMENT_ID	1 (null)	2 90	3 20	4 110		
DEPARTMENT_ID							
1 (null)							
2 90							
3 20							
4 110							

Στην περίπτωση που μετά το DISTINCT επιλέξουμε περισσότερες από μια στήλες τότε ο τελεστής λειτουργεί στο συνδυασμό των τιμών αυτών των στηλών.

```
SELECT DISTINCT department_id, job_id
FROM employees;
```

	DEPARTMENT_ID	JOB_ID
1	110	AC_ACCOUNT
2	90	AD_VP
3	50	ST_CLERK
4	80	SA_REP
5	50	ST_MAN

2.6 Ασκήσεις στο SELECT

1. Τα ακόλουθα SELECT εκτελούνται χωρίς σφάλματα;

```
SELECT last_name, job_id, salary AS Sal FROM employees; (ΣΩΣΤΟ/ΛΑΘΟΣ)
```

```
SELECT * FROM job_grades; (ΣΩΣΤΟ/ΛΑΘΟΣ)
```

2. Η παρακάτω εντολή περιέχει τέσσερα σφάλματα μπορείτε να τα εντοπίσετε;

```
SELECT employee_id, last_name sal x 12 ANNUAL SALARY FROM employees;
```

3. Μέσα από το pgAdmin ή τον SQL Developer δείτε την δομή και τα δεδομένα των πινάκων *DEPARTMENTS*, *EMPLOYEES*. Το τμήμα HR θέλει τα επώνυμα (last name) , των κωδικό θέσης (job code), την ημερομηνία πρόσληψης (hire date) και το αριθμό (employee number) του κάθε εργαζομένου με το πεδίο (στήλη *HIRE_DATE* να εμφανίζεται με το ψευδώνυμο *STARTDATE*). Σώστε το ερώτημα αυτό με το όνομα lab_01_05.sql
4. Το τμήμα HR θέλει μοναδικά του κωδικούς θέσεων από τον πίνακα *EMPLOYEES*.

5. Το τμήμα HR θέλει καλύτερες περιγραφές στο ερώτημα lab_01_05.sql για αυτό το λόγο δώστε κατάλληλα ονόματα στις στήλες *Emp#*, *Employee*, *Job* και *Hire Date* αντίστοιχα.
6. Το τμήμα HR ζητά επίσης μια αναφορά όπου θα φαίνονται όλοι οι υπάλληλοι και οι κωδικοί θέσεων. Εμφανίστε την αναφορά αυτή χρησιμοποιώντας το επίθετο σε συνένωση με τον κωδικό θέσης και ονομάστε τη στήλη *Employee and Title*
7. Εμφανίστε όλα τα δεδομένα του πίνακα *EMPLOYEES* χωρίζοντας τις τιμές όλων των πεδίων με κόμμα και ονομάστε την στήλη αυτή *THE_OUTPUT*.