



# Διάλεξη 12: SQL Data Definition Language

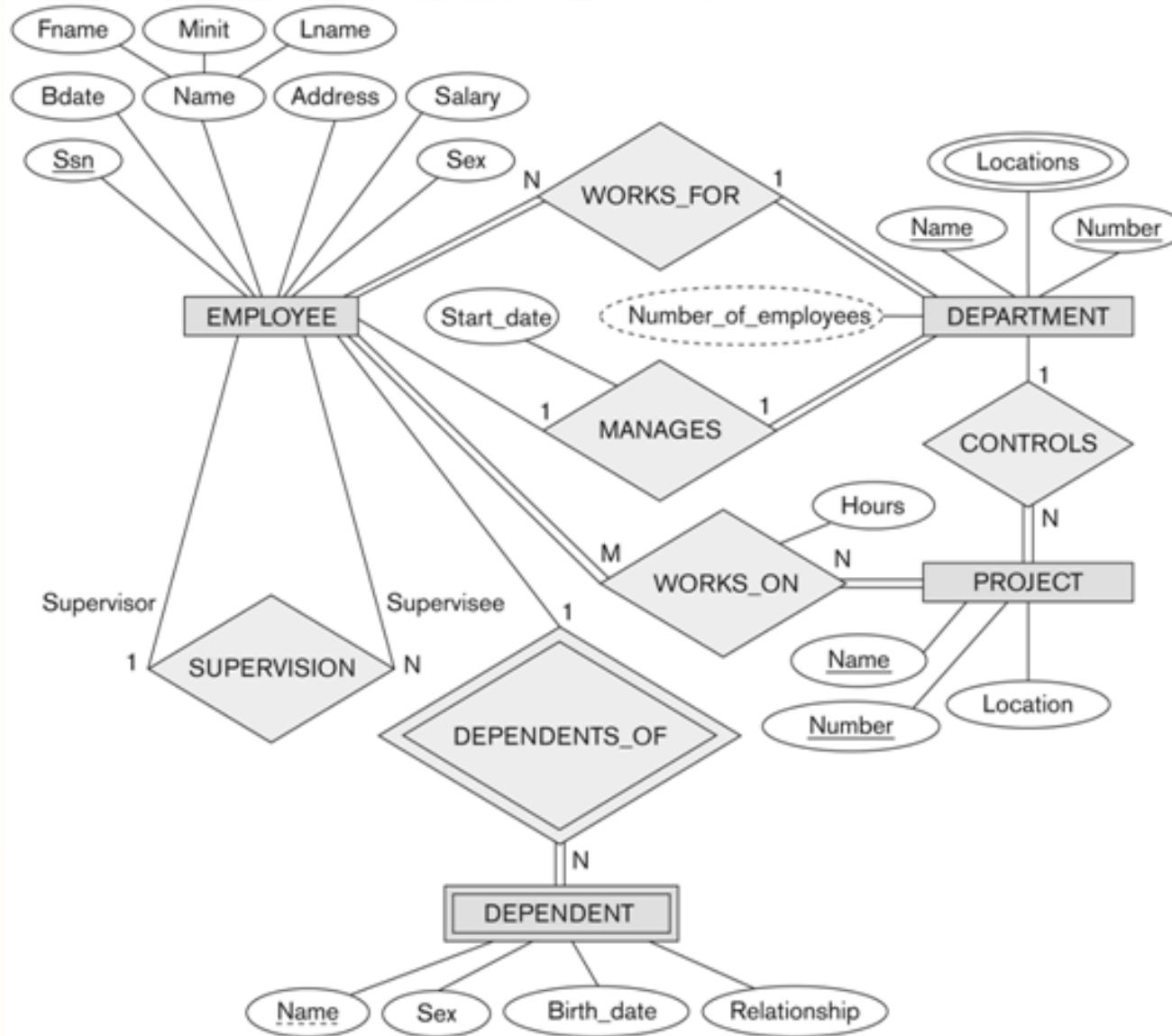
---

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:  
Εισαγωγή στις έννοιες:

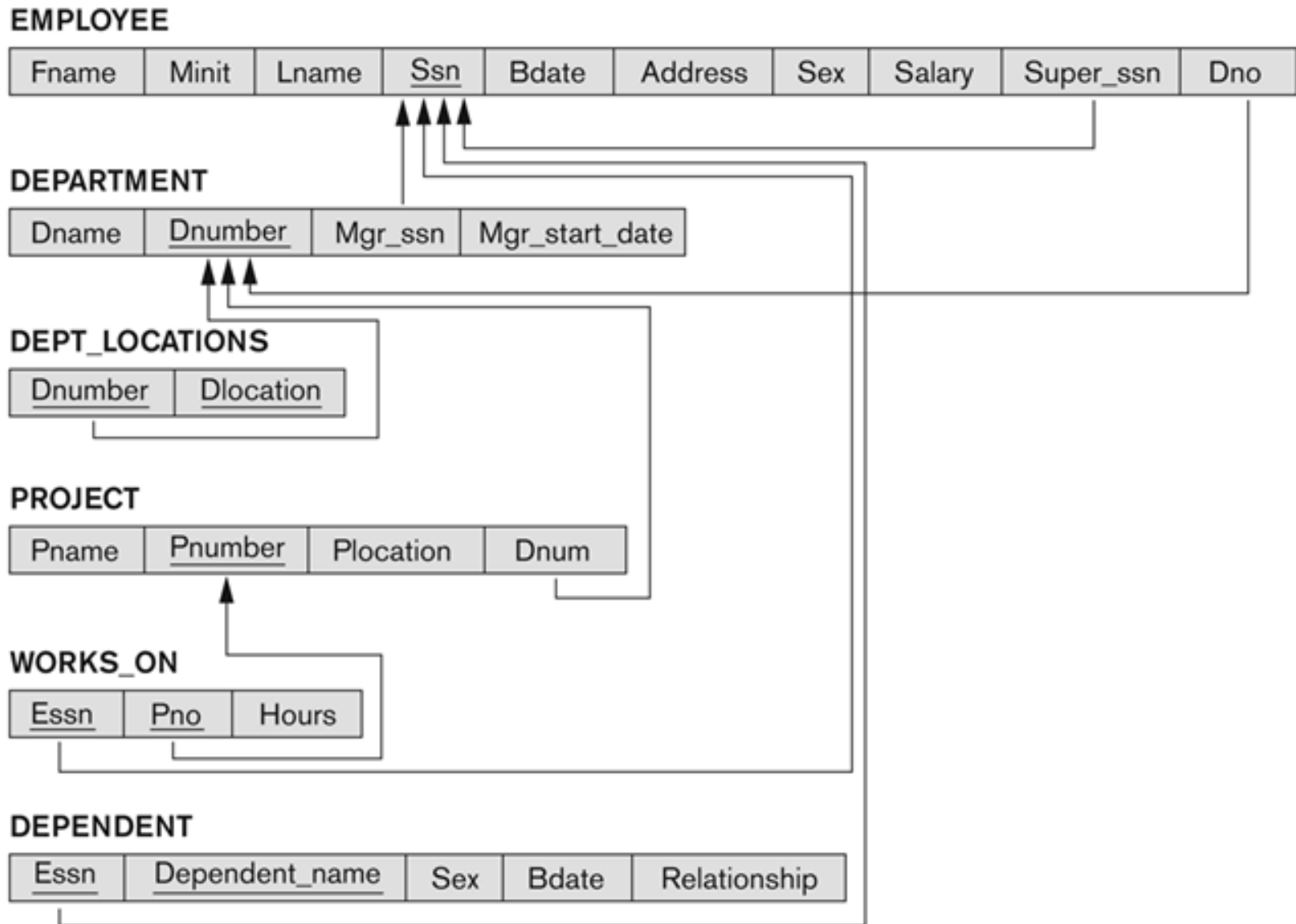
- Επιπρόσθετοι Περιορισμοί
- Παραδείγματα/Ασκήσεις

**Διδάσκων: Παναγιώτης Ανδρέου**

# Διάγραμμα ER



# Σχισιακό Σχήμα



# SQL-DDL

```
CREATE TABLE [dbo].[PROJECT](  
    [project_id] [int] NOT NULL,  
    [name] [nvarchar](50) NOT NULL,  
    [location] [nvarchar](50) NOT NULL,  
    [controlling_department] [int] NOT NULL  
    CONSTRAINT [PK_PROJECT] PRIMARY KEY  
    ([project_id] ASC)  
)
```

```
CREATE TABLE [dbo].[WORKS_ON](  
    [SSN] [int] NOT NULL,  
    [project_id] [int] NOT NULL  
    CONSTRAINT [PK_WORKS_ON] PRIMARY KEY  
    ([SSN] ASC, [project_id] ASC)  
)
```

```
CREATE TABLE [dbo].[DEPARTMENT](  
    [department_id] [int] NOT NULL,  
    [name] [nvarchar](50) NOT NULL,  
    [Manager] [int] NULL,  
    [Manager_start_date] [smalldatetime]
```

```
ALTER TABLE [dbo].[EMPLOYEE] WITH CHECK ADD  
CONSTRAINT [FK_EMPLOYEE_DEPARTMENT]  
FOREIGN KEY([Department])  
REFERENCES [dbo].[DEPARTMENT]  
([department_id])
```

```
ALTER TABLE [dbo].[EMPLOYEE] WITH CHECK ADD  
CONSTRAINT [DF_EMPLOYEE_ADDRESS] DEFAULT  
'N/A' FOR [Address]
```

```
ALTER TABLE [dbo].[DEPARTMENT] WITH CHECK  
ADD CONSTRAINT [UQ_DEPARTMENT_NAME]  
UNIQUE([name])
```

```
ALTER TABLE [dbo].[EMPLOYEE] WITH CHECK ADD  
CONSTRAINT [CHECK_EMPLOYEE_SALARY]  
CHECK([Salary]>=1000 AND [Salary]<=5000)
```

```
DROP TABLE [dbo].[DEPARTMENT]
```

...

# Επιπλέον Περιορισμοί στην SQL

- Θέματα Διάλεξης 11:
  - PRIMARY KEY
  - FOREIGN KEY
  - NOT NULL
- Θέματα Διάλεξης 12:
  - Ονομασία Περιορισμών
  - Επιπλέον περιορισμοί
    - UNIQUE
    - DEFAULT
    - CHECK
  - Εντολές Ενεργοποίησης Αναφοράς
    - ON DELETE | ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }
  - Προσωρινή Απενεργοποίηση Περιορισμών
    - WITH CHECK / WITH NO CHECK

# Ονομασία Περιορισμών

- Κάθε περιορισμός έχει κάποιο όνομα
- Όταν δεν δοθεί όνομα σε κάποιο περιορισμό από το χρήστη τότε η ΒΔ (DBMS) δίνει κάποιο τυχαίο όνομα
  - π.χ., PK\_EMPLOYEE\_123456
- Είναι πιο σωστό να ονομάζουμε τους περιορισμούς (όπου επιτρέπεται) ώστε να είναι πιο ευαναγνωστος ο κώδικας μας.
  - π.χ., PK\_EMPLOYEE, ή PRIMARY\_KEY\_EMPLOYEE
- Η ονομασία γίνεται κατά τον ορισμό του περιορισμού
  - π.χ., CREATE/ALTER ...  
**CONSTRAINT PK\_EMPLOYEE PRIMARY KEY EMPLOYEE(SSN)**



# UNIQUE Constraint

- **UNIQUE <γνωρίσματα>**: Προσδιορίζει ότι κάποιο γνώρισμα ή συνδυασμός γνωρισμάτων είναι μοναδικό
- **Μπορεί να είναι NULL** (βασική διαφορά από τα Primary Keys)
- Παράδειγμα:  

```
CREATE TABLE [dbo].[PROJECT](  
  [project_id] [int] NOT NULL,  
  [name] [nvarchar](50) UNIQUE,  
  [location] [nvarchar](50) NOT NULL,  
  ...)
```
- Όπως και για PKs, FKs όταν υπάρχει συνδυασμός πολλών γνωρισμάτων πρέπει να δηλωθεί στο τέλος του πίνακα ή με ALTER

# DEFAULT Constraint

- **DEFAULT <value>**: Ορίζει ένα **value** για ένα γνώρισμα δεδομένου ότι αυτό **δεν ορίζεται ρητά** κατά την εισαγωγή μιας πλειάδας σε μια σχέση.

- Παράδειγμα:

```
CREATE TABLE [dbo].[DEPARTMENT](
```

```
...
```

```
[Manager] [int] DEFAULT 0,
```

```
[Manager_start_date] [smalldatetime] DEFAULT GETDATE())
```

- Πληροφορίες για TSQL

- Το **value** μπορεί να είναι **NULL** ή **String** ή **scalar value**
- Το **DEFAULT** δεν εφαρμόζεται πάνω σε **IDENTITY**, **TIMESTAMP**



# CHECK Constraint

- **CHECK (logical expression):** Περιορισμός που επιβάλλεται στο πεδίο ορισμού ενός ή περισσότερων γνωρισμάτων
- Περιορίζει τις δυνατές τιμές ενός γνωρίσματος βάσει **λογικής συνθήκης**.
  - Η συνθήκη εφαρμόζεται σε κάθε αλλαγή (insert, update)
  - ΔΕΝ επιτρέπεται άμεση αναφορά σε άλλο πίνακα μέσω της CHECK. Επιτρέπεται έμμεσα με τη χρήση User Defined Function
- Παράδειγμα (γνωρίσματα του ίδιου πίνακα):

```
CREATE TABLE [dbo].[WORKS_ON](...  
    [HOURS] [int] CHECK (VALUE > 0),
```
- Παράδειγμα (γνωρίσματα άλλου πίνακα/ων):

```
CREATE TABLE [dbo].[DEPARTMENT](  
    ...  
    [NO_OF_EMPLOYEES] [int] CHECK (VALUE <= UDF_MAX_EMPLOYEES())
```
- Πληροφορίες για TSQL  
Δεν μπορεί να οριστεί για **text**, **ntext**, or **image**

# CHECK Constraint

- Παραδείγματα Λειτουργιών της **CHECK** σε **TSQL**

| Λειτουργία  | Σύνταξη σε TSQL  |
|---|--|
| Εξειδίκευση Πεδίου Ορισμού  | Month <b>BETWEEN</b> 1 and 12  |
| Εξειδίκευση Πεδίου Ορισμού  | Shippers <b>IN</b> ('Ups', 'Fed Ex', 'USPS')                                   |
| Σύγκριση Πεδίου με Σταθερά  | Price <b>&gt;=</b> 0   |
| Σύγκριση Πεδίων   | <b>DeliveryDate &gt;= OrderDate</b>  |
| Σύγκριση Πεδίου με Συνάρτηση  | StartDate <b>&lt;= GETDATE ( )</b>   |
| Έλεγχος Μορφής (e.g., 111-11-1111)<br>'[ ]': Οποιοσδήποτε χαρακτήρας μέσα<br>σε προσδιορισμένο εύρος (π.χ. [a-f]) ή<br>σύνολο ([abcdef]). | SSN <b>LIKE</b> '[0-9] [0-9] [0-9] -[0-9] [0-9] -<br>[0-9] [0-9] [0-9] [0-9] ' |
| Έλεγχος Μορφής<br>"%": 0 ή περισσότεροι χαρακτήρες<br>'_': 1 οποιοσδήποτε χαρακτήρας.   | email <b>LIKE</b> '%@cs.ucy.ac.cy'<br>'%@%.cy'                                 |

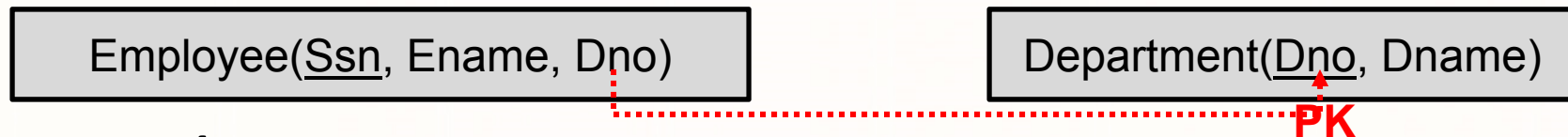
\* Το LIKE θα μελετηθεί εκτενέστερα στα πλαίσια της SQL-DML

# Εντολές Ενεργοποίησης Αναφοράς

- Οι **CHECK**, **NOT NULL**, **DEFAULT** είναι περιορισμοί που χρησιμοποιούνται στο επίπεδο **ενός γνωρίσματος** ή μιας **σχέσης**.
- Η **FOREIGN KEY** από την άλλη αναφέρεται σε περιορισμούς μεταξύ δυο σχέσεων.
  - π.χ., Π.χ., **FOREIGN KEY (MGRSSN) REFERENCES EMP(SSN)**
- Εάν παραβιαστεί ο αναφορικός περιορισμός τότε η βάση απορρίπτει την πράξη (**REJECT**)
  - Π.χ., δεν μπορούμε να σβήσουμε τον Manager κάποιου υπαλλήλου στην βάση.
- Για να εξειδικεύσουμε την αντίδραση της βάσης σε τέτοιες τροποποιήσεις χρησιμοποιούνται κάποιες επιπλέον εντολές στην δήλωση ενός πίνακα.
  - Δες επόμενη διαφάνεια

# Εντολές Ενεργοποίησης Αναφοράς

- Οι Εντολές Ενεργοποίησης Αναφοράς δηλώνουν τι θα γίνει στο **FK (Foreign Key)** μιας σχέσης από αλλαγή στο **PK (Primary Key)** αναφερόμενης σχέσης.
  - Ενεργοποιούνται αν παραβιαστεί ένας αναφορικός περιορισμός κατά το **ON DELETE** ή **ON UPDATE**



- **Λειτουργίες:**
  - **NO ACTION: Απαγορεύει την Αλλαγή (Default).**
    - Γιατί να ορίζεται; Για να ξεκαθαρίσει την πρόθεση του σχεδιαστή
  - **SET NULL: Θέτει το FK=NULL.**
    - Προϋποθέτει ότι η στήλη FK μπορεί να είναι NULLABLE.
  - **SET DEFAULT: Θέτει το FK=DEFAULT.**
    - Προϋποθέτει ότι το Default έχει οριστεί για το FK.
  - **CASCADE: Διάδοση Αλλαγής**
    - **ON DELETE CASCADE:** Διαγράφει την πλειάδα του FK
    - **ON UPDATE CASCADE:** Ενημερώνει το FK με το νέο PK

# Εντολές Ενεργοποίησης Αναφοράς

```
CREATE TABLE EMPLOYEE
(
  ...,
  Dno          INT          NOT NULL          DEFAULT 1,
  CONSTRAINT EMPCHK
  PRIMARY KEY(Ssn),
  CONSTRAINT EMPSUPERFK
  FOREIGN KEY(Super_ssn) REFERENCES EMPLOYEE(Ssn)
  ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT EMPDEPTFK
  FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber)
  ON DELETE SET DEFAULT ON UPDATE CASCADE );
CREATE TABLE DEPARTMENT
(
  ...,
  Mgr_ssn      CHAR(9)     NOT NULL          DEFAULT '888665555',
  ...,
  CONSTRAINT DEPTPK
  PRIMARY KEY(Dnumber),
  CONSTRAINT DEPTSK
  UNIQUE(Dname),
  CONSTRAINT DEPTMGRFK
  FOREIGN KEY(Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
  ON DELETE SET DEFAULT ON UPDATE CASCADE );
CREATE TABLE DEPT_LOCATIONS
(
  ...,
  PRIMARY KEY(Dnumber, Dlocation),
  FOREIGN KEY(Dnumber) REFERENCES DEPARTMENT(Dnumber)
  ON DELETE CASCADE ON UPDATE CASCADE );
```

Εάν διαγραφεί ο Supervisor του υπαλληλου X τότε X.Super\_ssn=NULL

Εάν αλλάξει SSN ο Supervisor του υπαλληλου X τότε X.Super\_ssn=NewSSN



# Εντολές Ενεργοποίησης Αναφοράς

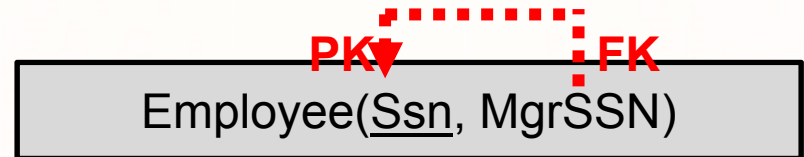
- Οι Εντολές Ενεργοποίησης Αναφοράς, ειδικότερα το **ON DELETE CASCADE**, χρειάζονται μεγάλη προσοχή διότι η ενέργεια δύναται να προκαλέσει **αλυσίδωτες αλλαγές (Chaining)** που μπορεί να σβήσει ολόκληρο πίνακα!

- Π.χ., Θεωρήστε την δήλωση

```
CREATE TABLE Employee (  
    SSN int NOT NULL PRIMARY KEY, MgrSSN int,  
    FOREIGN KEY(MgrSSN) REFERENCES Employee(SSN)  
    ON DELETE CASCADE)
```

Και το στιγμιότυπο `Employee = {(1,1), (2,1), (3,2), (4,3)}`

- Σβήνοντας τον `Employee.SSN=1` θα προκαλούσε διαγραφή όλων των εγγραφών του πίνακα!
- Για αυτό ο SQL Server δεν επιτρέπει την πιο πάνω εντολή `CREATE TABLE` επιστρέφοντας το μήνυμα λάθους”
  - Introducing FOREIGN KEY constraint 'FK\_\_Employee\_\_SSN\_\_1CF15040' on table 'Employee' may **cause cycles or multiple cascade paths**. Specify ON DELETE NO ACTION or ON UPDATE NO ACTION, or modify other FOREIGN KEY constraints.





# Προσωρινή Απενεργοποίηση Περιορισμών

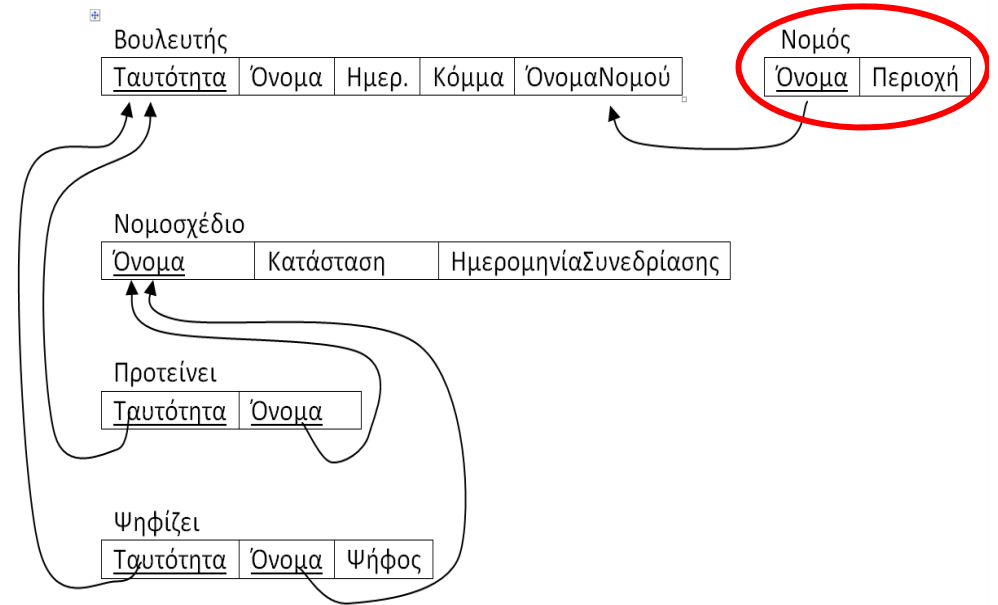
- Γνωρίζουμε ότι η βάση δεδομένων είναι συνεχώς σε μια **συνεπή (consistent)** κατάσταση.
    - Δηλαδή όλοι οι περιορισμοί (κλειδιού, αναφορικής ακεραιότητας, κτλ) τηρούνται με ακρίβεια.
  - Κάποτε επιθυμούμε να απενεργοποιήσουμε **προσωρινά** κάποιους περιορισμούς, π.χ.,
    - **Σειρά Εισαγωγής Δεδομένων:** εάν φορτώσουμε πρώτα τους Employee μετά τα Departments δεν θέλουμε να μας δίνει μηνύματα λάθους (λόγω foreign key constraint)
    - **Εισαγωγή Παλιών Δεδομένων,** θέλουμε να φορτώσουμε κάποια ΠΑΛΙΑ δεδομένα που δεν ακολουθούν κάποιο περιορισμό (π.χ., μορφοποίηση). Μπορούμε να τα φορτώσουμε και μετά να ενεργοποιήσουμε τον περιορισμό για τα νέα δεδομένα.
- ➔ ΔΕΝ ΣΥΝΙΣΤΑΤΑΙ ΤΟ ΣΕΝΑΡΙΟ 2 ΩΣΤΟΣΟ (ΑΣΥΝΕΠΕΙΑ)

# Προσωρινή Απενεργοποίηση Περιορισμών

- **WITH CHECK | WITH NOCHECK:** Δηλώνει κατά πόσο τα δεδομένα ενός πίνακα θα επικυρωθούν (validated) ή όχι έναντι κάποιου περιορισμού (π.χ., περιορισμό FOREIGN KEY ή CHECK)
- **Επισημάνσεις**
  - Οι περιορισμοί πρέπει να απενεργοποιούνται **ΠΡΟΣΩΡΙΝΑ**.
  - Δεν μπορούμε να απενεργοποιήσουμε το **PRIMARY KEY** και **UNIQUE**
  - Ενεργοποίηση/Απενεργοποίηση θα γίνει με χρήση της εντολής **ALTER**.
- **Παραδείγματα**
  - **Ενεργ./Απενεργ. Υφιστάμενου Περιορισμού:**  
ALTER TABLE Employee **CHECK | NOCHECK** CONSTRAINT fk\_something
  - **Εισαγωγή Νέου Περιορισμού FK (χωρίς έλεγχο παλιών data): :**  
ALTER TABLE Employee **WITH NOCHECK** → WITH CHECK είναι default για νέους περιορισμούς  
**ADD CONSTRAINT FK\_Employee FOREIGN KEY(SSN) REFERENCES Employee(SSN)**
  - **Εισαγωγή Νέου Περιορισμού CHECK (χωρίς έλεγχο παλιών data):**  
ALTER TABLE Employee **WITH NOCHECK**  
**ADD CONSTRAINT CN\_SSNFormat CHECK (SSN LIKE**  
**'[0-9][0-9][0-9][0-9][0-9][0-9][0-9]')**

# Ασκήσεις RM → SQL-DDL

- Δώστε SQL-DDL (CREATE) για:
  - Σχέση Νομός



```
CREATE TABLE [Νομός](  
    [Όνομα] [nvarchar](50) NOT NULL,  
    [Περιοχή] [nvarchar](50),
```

```
    PRIMARY KEY (Όνομα),
```

```
    CONSTRAINT PK_Νομός  
)
```

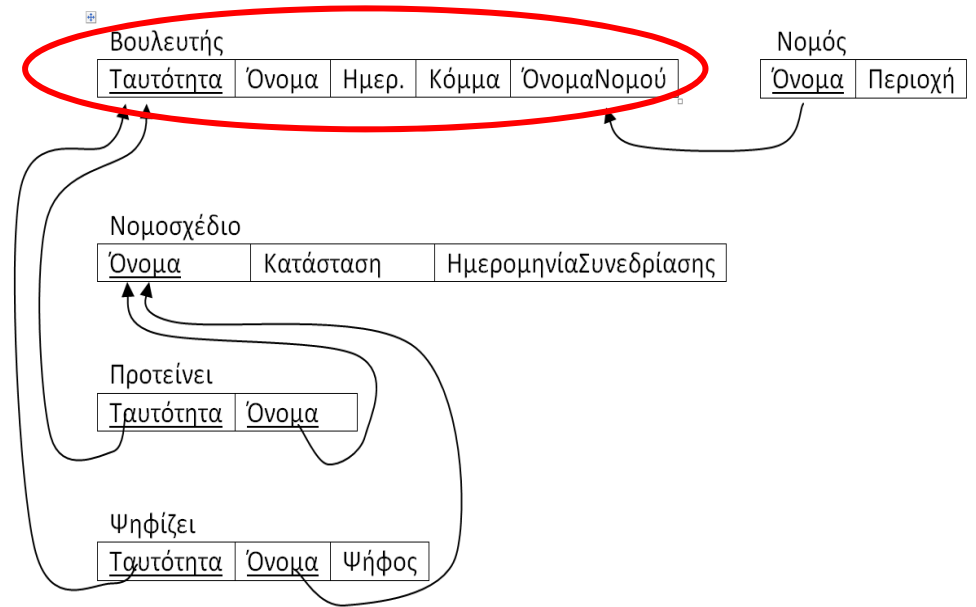
# Ασκήσεις RM → SQL-DDL

- Δώστε SQL-DDL (CREATE) για:
  - Σχέση Βουλευτής

```
CREATE TABLE [Βουλευτής](  
  [Ταυτότητα] [int] NOT NULL,  
  [Όνομα] [nvarchar](50),  
  [Ημερ.] [smalldatetime],  
  [Κόμμα] [nvarchar](50),  
  [O_Νομού] [nvarchar](50),
```

```
  CONSTRAINT PK_Βουλευτής  
  CONSTRAINT FK_Βουλευτής_Νομός
```

)

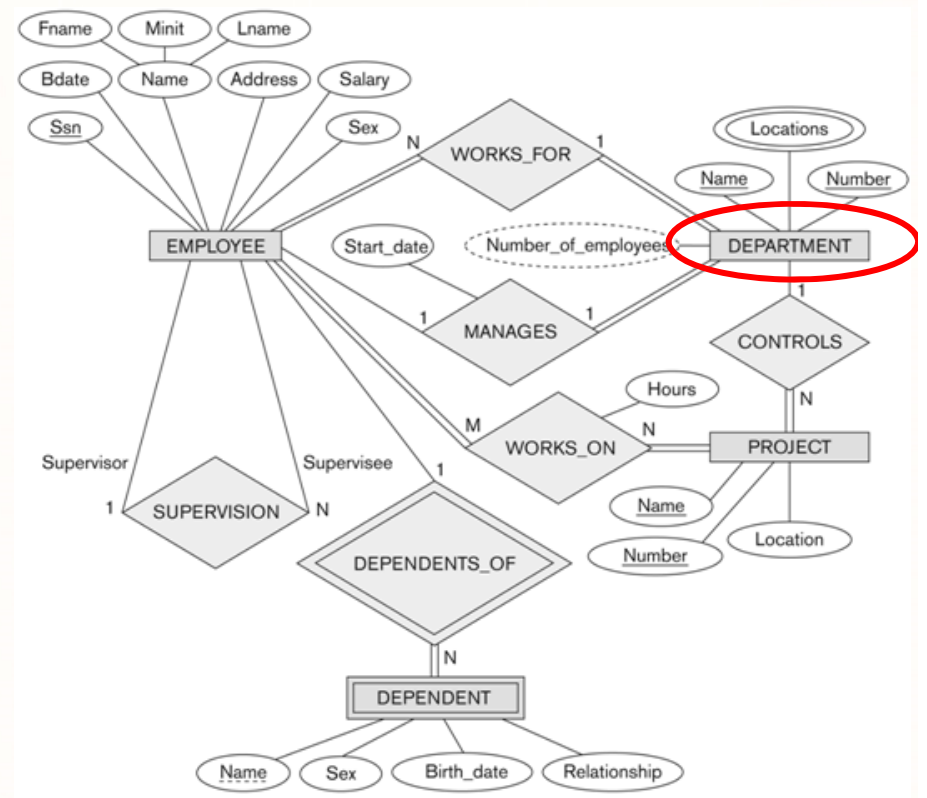


```
PRIMARY KEY (Ταυτότητα),  
FOREIGN KEY (O_Νομού)  
REFERENCES Νομός (Όνομα)
```

# Ασκήσεις ER → SQL-DDL

- Υποθέστε:
  - Employee (SSN)
- Δώστε SQL-DDL (CREATE) για:
  - Οντότητα DEPARTMENT

```
CREATE TABLE [DEPARTMENT](  
  [Dnumber] [int] NOT NULL,  
  [Dname] [nvarchar](50) NOT NULL,  
  [Manager] [int] NOT NULL,  
  [MgrStartD] [smalldatetime],
```



```
CONSTRAINT PK_DEPARTMENT  
CONSTRAINT UQ_DEPARTMENT_NAME  
CONSTRAINT FK_DEPARTMENT_MANAGER
```

```
PRIMARY KEY (Dnumber),  
UNIQUE (Dname),  
FOREIGN KEY (Manager)  
REFERENCES EMPLOYEE (SSN)
```

)

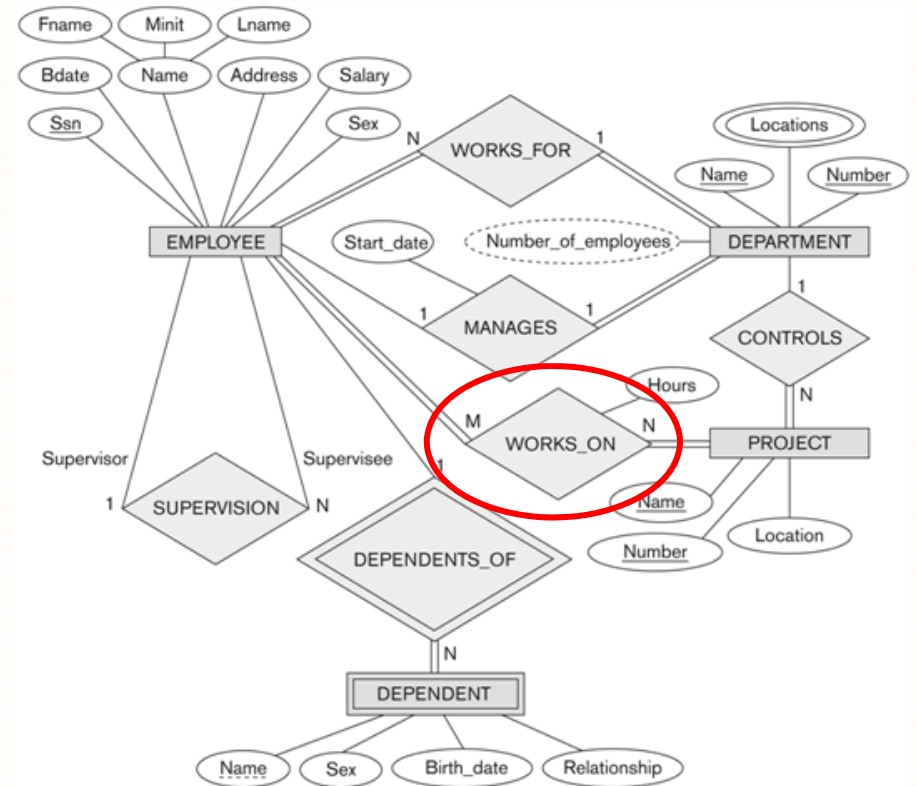


# Ασκήσεις ER → SQL-DDL

- Υποθέστε:
  - Employee (SSN)
  - PROJECT (Pnumber)
- Δώστε SQL-DDL (CREATE) για:
  - Συσχέτιση WORKS\_ON

```
CREATE TABLE [WORKS_ON](  
  [Pnumber] [int] NOT NULL,  
  [SSN] [int] NOT NULL,  
  [PHours] [int] NULL,  
  
  CONSTRAINT PK_WORKS_ON  
  CONSTRAINT FK_WORKS_ON_PROJECT  
  CONSTRAINT FK_WORKS_ON_EMPLOYEE
```

```
PRIMARY KEY (Pnumber, SSN),  
FOREIGN KEY (Pnumber)  
REFERENCES PROJECT (Pnumber)  
FOREIGN KEY (SSN)  
REFERENCES EMPLOYEE (SSN)
```





# Ασκήσεις ER → SQL-DDL

- Υποθέστε:

- Employee (SSN, ...)
- Συσχέτιση Supervision αναγνωρίστηκε μετά από τη δημιουργία του πίνακα

- Δώστε SQL-DDL (ALTER) για:

- Συσχέτιση SUPERVISION

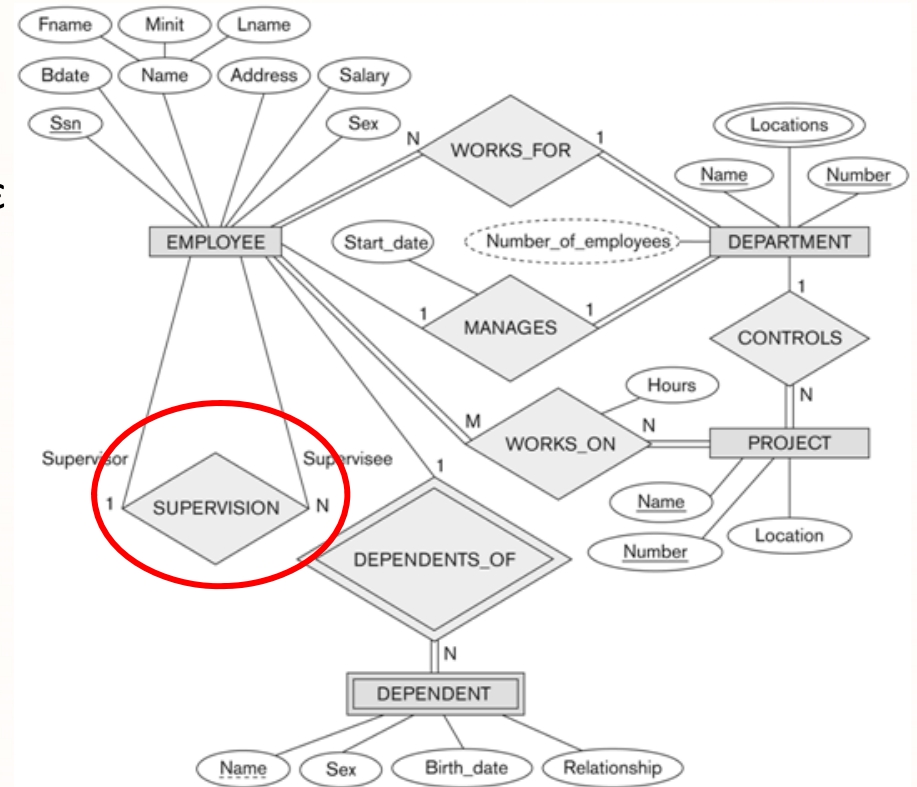
```
ALTER TABLE [dbo].[EMPLOYEE]
```

```
ADD Supervisor [int] NULL
```

```
ALTER TABLE [dbo].[EMPLOYEE]
```

```
ADD CONSTRAINT FK_EMPLOYEE_SUPER
```

**FOREIGN KEY (Supervisor)  
REFERENCES EMPLOYEE (SSN)**



# Ασκήσεις ER → SQL-DDL

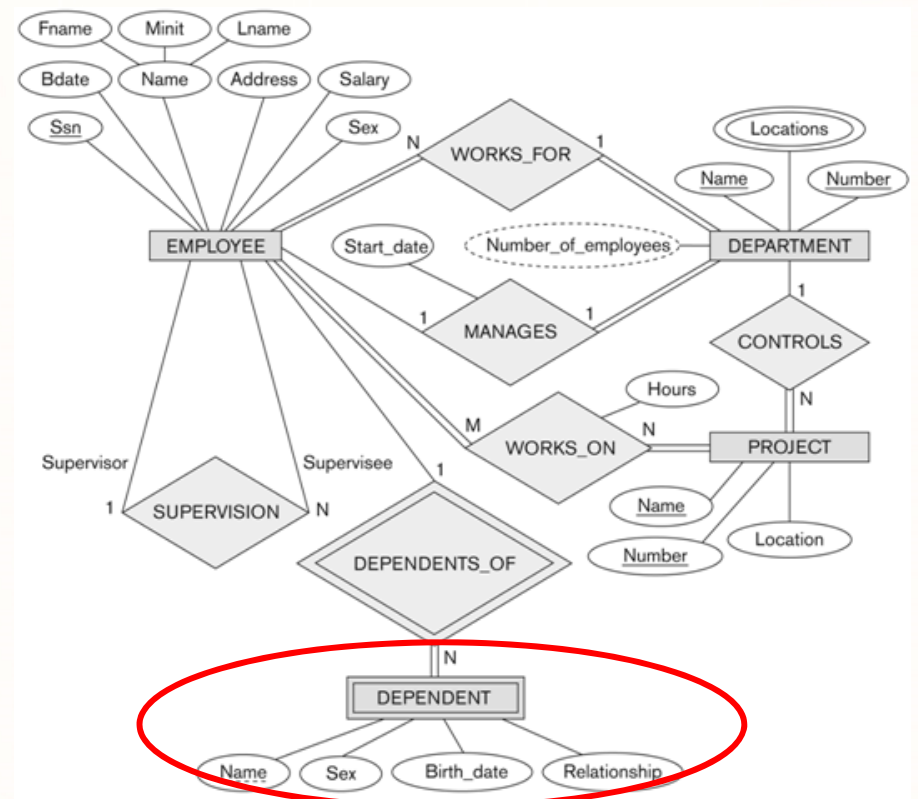
- Υποθέστε:

- Employee (SSN)

- Δώστε SQL-DDL (CREATE) για:

- Οντότητα DEPENDENT

```
CREATE TABLE [DEPENDENT](
  [ESSN] [int] NOT NULL,
  [Name] [nvarchar](50) NOT NULL,
  [Sex] [bit],
  [DOB] [smalldatetime],
  [Relationship] [nvarchar](50),
```



**PRIMARY KEY** (ESSN, Name),  
**FOREIGN KEY** (ESSN)  
**REFERENCES** EMPLOYEE (SSN)

)

# Ασκήσεις RM → SQL-DDL

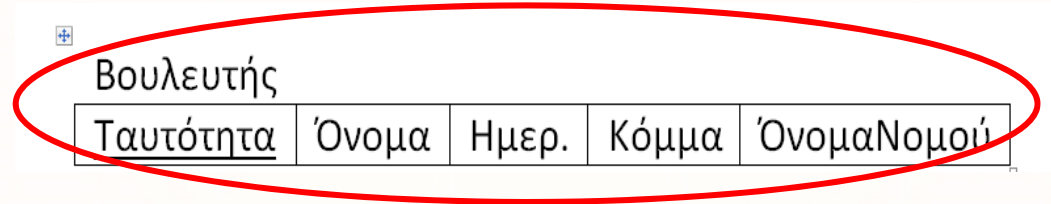
- Υποθέστε:

- Σχέση Βουλευτής

... [Όνομα] [nvarchar](50),

[Ημερ.] [smalldatetime],

[Κόμμα] [nvarchar](50) CHECK ( VALUE IN (Κόμμα-Α, Κόμμα-Β, Κόμμα-Γ) )



| Βουλευτής |       |       |       |            |
|-----------|-------|-------|-------|------------|
| Ταυτότητα | Όνομα | Ημερ. | Κόμμα | ΌνομαΝομού |

- Δώστε SQL-DDL (ALTER) για τους ακόλουθους περιορισμούς:

- Το μέγεθος του ονόματος να είναι μεγαλύτερο από 6 χαρακτήρες (LEN())

```
ALTER TABLE [Βουλευτής] ADD CONSTRAINT CH_CHECK_LENGTH_Όνομα  
CHECK ( LEN([Όνομα]) > 6 )
```

- Το κόμμα να είναι από τη λίστα (Κόμμα-Α, Κόμμα-Β, Κόμμα-Γ)

```
ALTER TABLE [Βουλευτής] ADD CONSTRAINT CH_CHECK_Κόμμα  
CHECK ( [Κόμμα] IN (Κόμμα-Α, Κόμμα-Β, Κόμμα-Γ) )
```

- Η Ημερ. να είναι μεγαλύτερη της ημερομηνίας καταχώρησης

```
ALTER TABLE [Βουλευτής] ADD CONSTRAINT CH_CHECK_Ημερ_Καταχώρησης  
CHECK ( Ημερ.>= GETDATE() )
```

# Ασκήσεις ER → SQL-DDL

- Υποθέστε:
  - Σχέση Άτομο ...
    - [Όνομα] [nvarchar](50) NOT NULL,
    - [Επίθετο] [nvarchar](50) NOT NULL,
    - ...
- Δώστε SQL-DDL (ALTER) για τους ακόλουθους περιορισμούς:
  - Δημιουργία υπολογιζόμενου γνωρίσματος [Ονοματεπώνυμο]

```
ALTER TABLE [Άτομο]
ADD [Ονοματεπώνυμο] AS
[Όνομα] + ' ' + [Επίθετο]
```

# Ασκήσεις ER → SQL-DDL

- Υποθέστε:

- Σχέση Άτομο ...

|                |                |              |
|----------------|----------------|--------------|
| [Όνομα]        | [nvarchar](50) | NOT NULL,    |
| [Μεσαίο Όνομα] | [nvarchar](1)  | <b>NULL,</b> |
| [Επίθετο]      | [nvarchar](50) | NOT NULL,    |

...

- Δώστε SQL-DDL (ALTER) για τους ακόλουθους περιορισμούς:

- Δημιουργία υπολογιζόμενου γνωρίσματος [ΠλήρεςΌνομα]  
*ΣΗΜΕΙΩΣΗ: Η Συνένωση συμβολοσειράς με NULL → NULL*

```
ALTER TABLE [Άτομο]
```

```
ADD [ΠλήρεςΌνομα] AS
```

```
[Όνομα] + ISNULL(' ' + [Μεσαίο Όνομα] + ' ', ' ') + [Επίθετο]
```