



Databases and XML (2) – 15.02.2016

Time	Activity
08.30	Intro and some code examples from last time
08.40	Crow's foot vs Chen vs UML notation
08.50	ER exercise
09.10	Normalization and Views
09.20	SQL assignment (focus on getting help on SQL-2)
10.00	Break
10.30	JOIN (Venn diagrams)
10.40	JOIN exercise
11.10	SQL assignment (focus on getting started at SQL-3)
11.50	Homework and next lecture

Today's agenda



Datamodel 1:

Assignment3PollApp...p (EFIF\task (54))

```
----- QUESTION 4 -----  
  
SELECT COUNT(U.Email) as CountOption, [Option]  
FROM dbo.[Option] as O  
LEFT OUTER JOIN dbo.Answer as A  
    ON A.OptionId = O.OptionId  
    LEFT OUTER JOIN dbo.[User] as U  
        ON U.UserId = A.UserId  
WHERE O.QuestionId = 2  
GROUP BY O.[Option];
```

100 %

Results Messages

	CountOption	Option
1	1	Banana cake
2	2	Chokolade cake
3	0	Cucumber muffin

Assignment 3.4

Datamodel 2:

```
--- 4 ---
USE PollApp2;
SELECT count(U.Email) as CountOption, O.[Option]
FROM dbo.OptionQuestion as OQ
    LEFT OUTER JOIN dbo.[Option] as O ON OQ.OptionId = O.OptionId
    LEFT OUTER JOIN dbo.Answer as A ON A.OptionId = O.OptionId
    LEFT OUTER JOIN dbo.[User] as U ON U.UserId = A.UserId
WHERE OQ.QuestionId = 1
GROUP BY O.[Option]

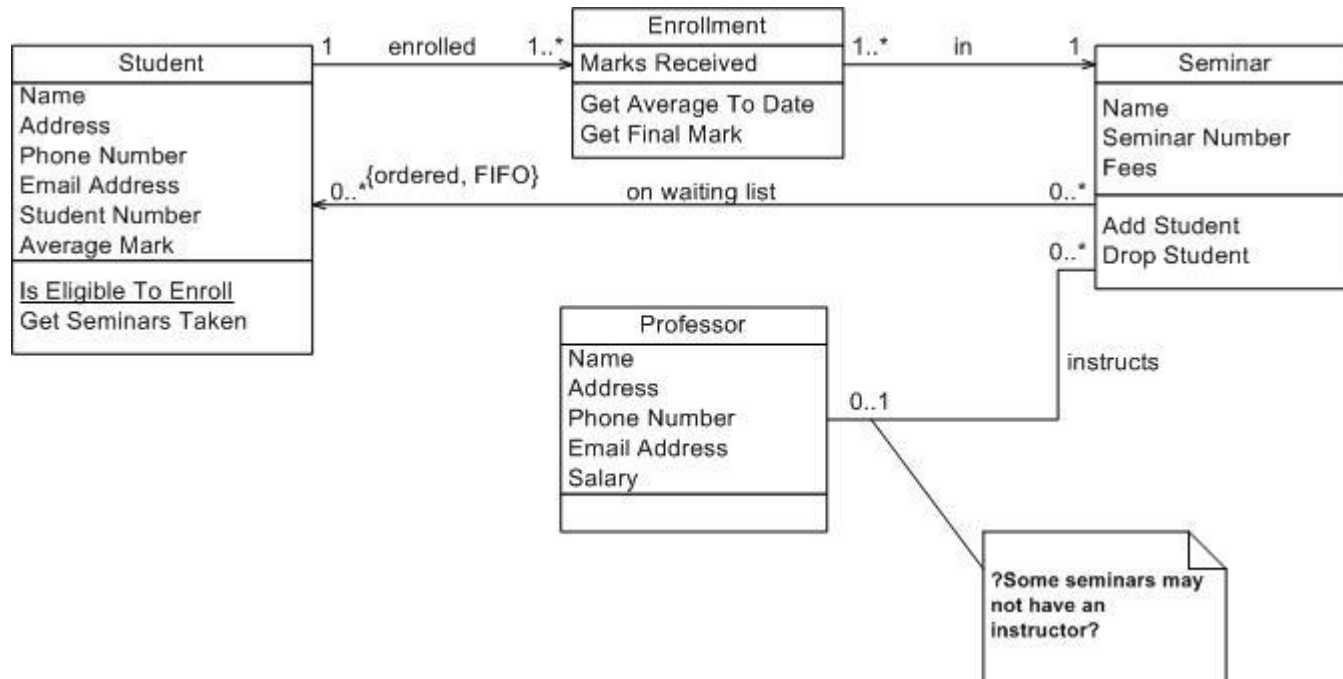
--- THE SIMPLER SOLUTION ---

SELECT COUNT(A.OptionId) AS CountOption, [Option]
FROM [Option] AS O
    LEFT JOIN Answer AS A ON A.OptionId = O.OptionId
    JOIN OptionQuestion AS OQ ON OQ.OptionId = O.OptionId
WHERE OQ.QuestionId = 1
GROUP BY O.[Option]
```

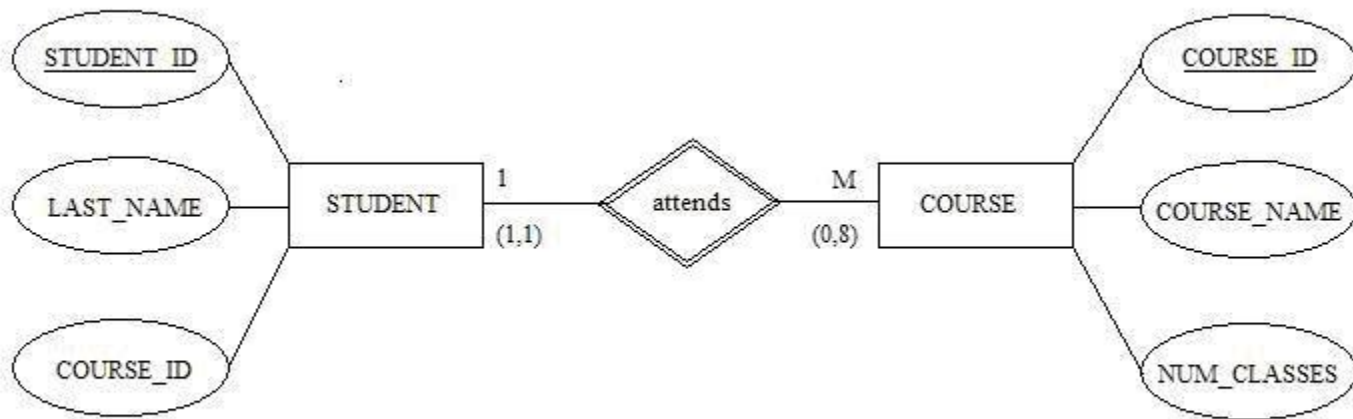
	CountOption	Option
1	1	Cat
2	1	Dog
3	1	Dolphin
4	0	Elephant

	CountOption	Option
1	1	Cat
2	1	Dog
3	1	Dolphin
4	0	Elephant

Assignment 3.4



UML diagram



Student = Instructor/teacher

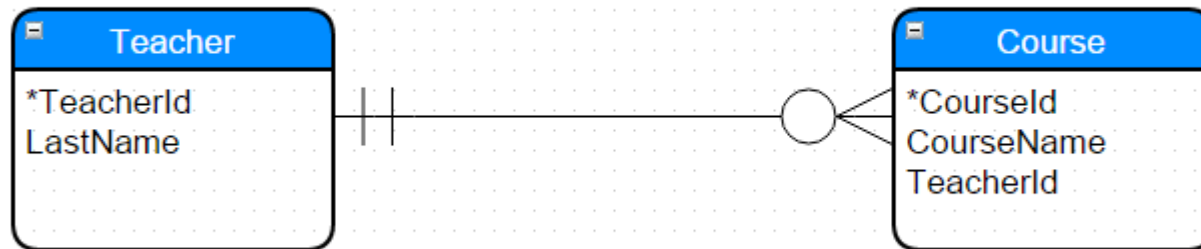
Chen's notation

- Weak relationship – the foreign key is part of the primary key
- A weak relationship is non-identifying
- A weak entity is an entity whose primary key includes the primary key of a related entity
- Weak entities depend on the existence of their related entities. If you delete the related entity, you must cascade that deletion to the weak entities that contain the primary key you've deleted
- More relevant in the object world

- Chen: double diamond
- Crow: strong line (in stead of the dotted line)

Weak relationship





Crow's foot notation

ER exercise



1. Describe the underlying database design

```
SELECT Productgroup, Price, Product
FROM Product AS P
INNER JOIN Productgroup AS PG ON pg.ProductgroupId = p.ProductgroupId
ORDER BY Product
```

2. What SQL expression can create this dataset?

	TotalValue	Productgroup
1	5000.00	Computer
2	20.00	Keyboard

ER exercise

Continue SQL assignment



1NF means the tuples (rows) in the relation (table) must be unique.



2NF means every non-key attribute has to be fully functionally dependent on the entire candidate key.

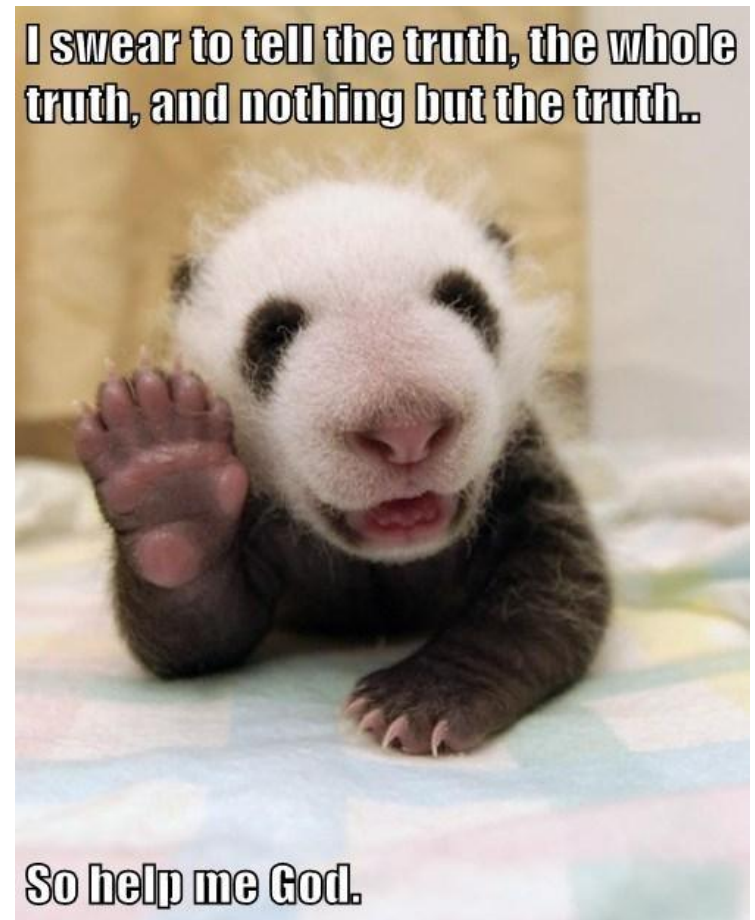


3NF means that all non-key attributes must be mutually independent.

Normalization



Every non-key attribute is dependent on the key, the whole key, and nothing but the key – so help me Codd.



Informally for 2NF and 3NF

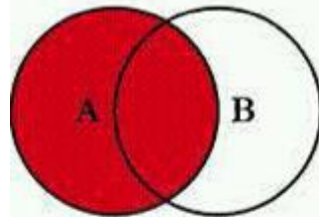
Normalization example SQL

- Simple database object
- Creates a virtual table
- Encourages reusing code, better testing and fewer bugs
- Can be used as a security mechanism
- Consists of a SELECT statement
 - It can be a single SELECT query that is saved with a name in the database
 - Can also be a complex query – where the complexity is hidden for users/web developers/software programmers
- Cannot take parameters
- Avoid using SELECT * in Views

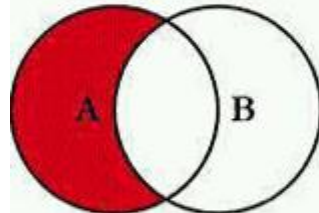
Introduction to Views



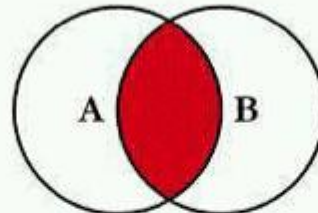
SQL JOINS



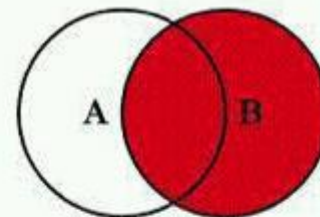
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key



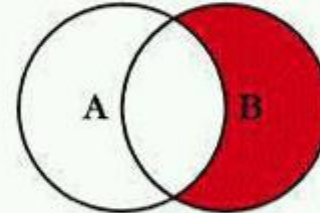
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL



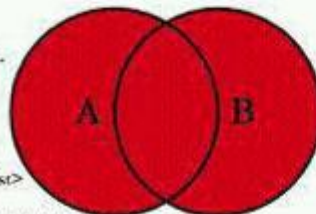
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key



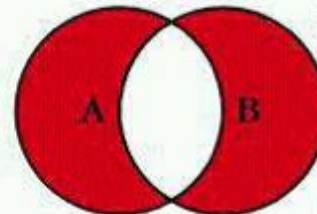
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key



SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL



SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key



SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL

© C.L. McFatt, 2008

JOIN (Venn diagrams)

Exercise



Group 1: Left join

Group 2: Right join

Group 3: Inner join

Group 4: Left join where B.key IS NULL

Group 5: Right join where A.key IS NULL

Group 6: Full outer join

Group 7: Full outer join where A.key IS NULL OR B.key IS NULL

Make a document where you explain the join you were assigned, and most importantly make an example of your own to illustrate it.

Group exercise



Continue SQL assignment



Next week's topic:

SQL: Stored procedures and Functions

Exercises – finish:

- SQL-1 – poll app (from lesson 01)
- SQL-2 – Start of the blog (from lesson 01)
- SQL-3 – Date and recursive categories (from lesson 02)

Read:

- T-SQL pages 362-370
- Beginning T-SQL – Stored Procedures
- SQL Server for Dummies

Pluralsight:

- No new video to watch this week
- Take this opportunity to rewatch the two previous videos

Homework and preparation

