

12. Функционални зависимости

Лекционен курс "Бази от данни"

Въведение

- С тази лекция започваме въведение в проектирането на бази данни;
- Основно две взаимосвързани концепции:
 - Функционални зависимости
 - Теория на нормализацията

Основни понятия

Функционалните зависимости са взаимоотношения между атрибути в една релация.

Една ФЗ (FD) е едно много-към-едно отношение на едно множество от атрибути към друго множество от атрибути в **една релация**.

С други думи, ако знаем стойността на една комбинация от полета от определен запис, то можем да намерим стойностите на други полета от същия запис.

Напр.,:

- Ако знаем ЕГН на определен човек, това значи, че можем да определим и името на този човек;
- За да разберем каква оценка е получил даден студент, трябва да знаем кой е студентът и коя е дисциплината, оценката от която искаме да научим.

В релацията LECTURER има функционална зависимост между {ID} и {FNAME}. Това значи, че за всяка конкретна стойност на атрибута ID има точно една кореспондираща стойност на атрибута FNAME.

LECTURER

ID	FNAME	LNAME	TITLE	SALARY
101	Стоян	Колев	Проф.	800
102	Петър	Иванов	Доц.	600
103	Венета	Георгиева	Ас.	400

RATING

FAC_NO	LECTION	ASSESSMENT
S1	DB	5
S1	AI	3
S2	DB	4
S2	SE	6
S1	SE	5

В релацията RATING има функционална зависимост между {FACNO, LLECTION} и {ASSESSMENT}, т.е. студентът и лекцията определят точно една кореспондираща оценка на този студент за тази лекция.

Дефиниция - 1-ви случай:

Нека:

- R – релация;
- X, Y – произволни подмножества на множеството на атрибутите на R.

Тогава:

- $X \rightarrow Y$: “X функционално определя Y” или “Y функционално зависи от X”, ако и само ако всяка X-стойност в R е свързана точно с една Y-стойност в R.

Примери:

- $ЕГН \rightarrow \text{име_човек}$
- $(\text{студент, дисциплина}) \rightarrow \text{оценка}$

Атрибутите от лявата страна се наричат детерминанти.

Ако детерминантът се състои от повече от един атрибут се нарича съставен.

Накратко:

- Функционалните зависимости са взаимоотношения между стойности на атрибути;
- Напр., ако ЕГН определя еднозначно един човек, то на този номер ще съответства точно едно име на човек.

Много съществено е да различаваме:

- Стойност на една релация в точно определен момент;
- Множеството от всички възможни стойности, които релацията може да има в различни моменти.

STUDENTS

FNO	YEAR	SUBJ	ASS
021039	Inf-III	DB	4
021039	Inf-III	AI	4
021142	Math-III	DB	5
021142	Math-III	AI	5
021144	Math-III	AI	6
021001	Inf-III	AI	3
021001	Inf-III	SE	3
021001	Inf-III	Java	3



STUDENTS

FNO	YEAR	SUBJ	ASS
021039	Inf-III	DB	4
021039	Inf-III	AI	4
021142	Math-III	DB	5
021142	Math-III	AI	5
021144	Math-III	AI	6
021001	Inf-III	AI	3
021001	Inf-III	SE	3
021001	Inf-III	Java	3

Функционални зависимости:

- $\{ FNO \} \rightarrow \{ YEAR \}$
- $\{ FNO, SUBJ \} \rightarrow \{ ASS \}$
- $\{ FNO, SUBJ \} \rightarrow \{ YEAR \}$
- $\{ FNO, SUBJ \} \rightarrow \{ YEAR, ASS \}$
- $\{ FNO, SUBJ \} \rightarrow \{ FNO \}$
- $\{ FNO, SUBJ \} \rightarrow \{ FNO, SUBJ, YEAR, ASS \}$
- $\{ FNO \} \rightarrow \{ ASS \}$
- $\{ ASS \} \rightarrow \{ FNO \}$



Дефиниция - 2-ри случай:

Нека:

- R – реляционна променлива;
- X, Y – произволни подмножества на множеството на атрибутите на R .

Тогава:

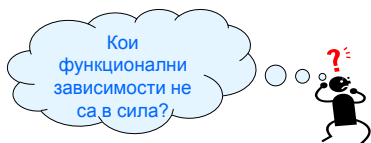
- $X \rightarrow Y$ – “ Y функционално зависи от X ” или “ X функционално определя Y ”, ако и само ако **във всяка възможна** коректна стойност на R , всяка X -стойност е свързана точно с една Y -стойност.

STUDENTS

FNO	YEAR	SUBJ	ASS
021039	Inf-III	DB	4
021039	Inf-III	AI	4
021142	Math-III	DB	5
021142	Math-III	AI	5
021144	Math-III	AI	6
021001	Inf-III	AI	3
021001	Inf-III	SE	3
021001	Inf-III	Java	3

Функционални зависимости:

- $\{FNO\} \rightarrow \{YEAR\}$
- $\{FNO, SUBJ\} \rightarrow \{ASS\}$
- $\{FNO, SUBJ\} \rightarrow \{YEAR\}$
- $\{FNO, SUBJ\} \rightarrow \{YEAR, ASS\}$
- $\{FNO, SUBJ\} \rightarrow \{FNO\}$
- $\{FNO, SUBJ\} \rightarrow \{FNO, SUBJ, YEAR, ASS\}$
- $\{FNO\} \rightarrow \{ASS\}$
- $\{ASS\} \rightarrow \{FNO\}$



STUDENTS

FNO	YEAR	SUBJ	ASS
021039	Inf-III	DB	4
021039	Inf-III	AI	4
021142	Math-III	DB	5
021142	Math-III	AI	5
021144	Math-III	AI	6
021001	Inf-III	AI	3
021001	Inf-III	SE	3
021001	Inf-III	Java	3
021001	Inf-III	DB	4

Функционални зависимости:

- $\{FNO\} \rightarrow \{YEAR\}$
- $\{FNO, SUBJ\} \rightarrow \{ASS\}$
- $\{FNO, SUBJ\} \rightarrow \{YEAR\}$
- $\{FNO, SUBJ\} \rightarrow \{YEAR, ASS\}$
- $\{FNO, SUBJ\} \rightarrow \{FNO\}$
- $\{FNO, SUBJ\} \rightarrow \{FNO, SUBJ, YEAR, ASS\}$
- ~~$\{FNO\} \rightarrow \{ASS\}$~~
- ~~$\{ASS\} \rightarrow \{FNO\}$~~



Пример

Нека разгледаме следния пример – релацията, съдържаща доставчици и части.

S	CITY	P	QTY
S1	Пловдив	P1	100
S1	Пловдив	P2	100
S2	Смолян	P1	200
S2	Смолян	P2	200
S3	Смолян	P2	300
S4	Пловдив	P2	400
S4	Пловдив	P4	400
S4	Пловдив	P5	400

$\{S, P\} \rightarrow \{QTY\}$
 $\{S, P\} \rightarrow \{CITY\}$
 $\{S, P\} \rightarrow \{CITY, QTY\}$
 $\{S, P\} \rightarrow \{S\}$
 $\{S, P\} \rightarrow \{S, P, CITY, QTY\}$
 $\{S\} \rightarrow \{QTY\}$
 $\{QTY\} \rightarrow \{S\}$

Пример

S#	CITY	P#	QTY
S1	Пловдив	P1	100
S1	Пловдив	P2	100
S2	Смолян	P1	200
S2	Смолян	P2	200
S3	Смолян	P2	300
S4	Пловдив	P2	400
S4	Пловдив	P4	400
S4	Пловдив	P5	400

$\{S\#, P\# \} \rightarrow \{QTY\}$
 $\{S\#, P\# \} \rightarrow \{CITY\}$
 $\{S\#, P\# \} \rightarrow \{CITY, QTY\}$
 $\{S\#, P\# \} \rightarrow \{S\# \}$
 $\{S\#, P\# \} \rightarrow \{S\#, P\#, CITY, QTY\}$
 $\{S\# \} \rightarrow \{QTY\}$
 $\{QTY\} \rightarrow \{S\# \}$

Тук можем да забележим, че следните функционални зависимости **не са** в сила за всяка възможна стойност на релационната променлива:

$S\# \rightarrow QTY$

$QTY \rightarrow S\#$

S#	CITY	P#	QTY
S1	Пловдив	P1	100
S1	Пловдив	P2	100
S2	Смолян	P1	200
S2	Смолян	P2	200
S3	Смолян	P2	300
S4	Пловдив	P2	400
S4	Пловдив	P4	400
S4	Пловдив	P5	400
S4	Пловдив	P3	200

$\{S\#, P\# \} \rightarrow \{QTY\}$
 $\{S\#, P\# \} \rightarrow \{CITY\}$
 $\{S\#, P\# \} \rightarrow \{CITY, QTY\}$
 $\{S\#, P\# \} \rightarrow \{S\# \}$
 $\{S\#, P\# \} \rightarrow \{S\#, P\#, CITY, QTY\}$
 $\{S\# \} \rightarrow \{QTY\}$
 $\{QTY\} \rightarrow \{S\# \}$

Т.е. твърдението, че "всяка доставка за конкретен доставчик съдържа едно и също количество" не е вярно за всички възможни стойности на релацията.

Съществен извод:

Всички атрибути в една релация задължително трябва да зависят от първичния ключ!

Ако това условие не е изпълнено, тогава релацията съдържа някакво излишество.

- Както видяхме, дори при малкия пример множеството на функционалните зависимости беше доста голямо;
- Основен въпрос – може ли да бъде редуцирано?

Защо е толкова важно множеството на ФЗ да бъде редуцирано?

- Една причина е, че ФЗ представят ограничения за цялостност, които БД трябва да проверява при промяна на данните;
- Така че за дадено множество от ФЗ S ако можем да намерим друго T , много по-малко от него, такова, че всяка ФЗ от S може да се изведе от тези от T , ще е достатъчно БД да се съобразява само с тези ФЗ от T , като така всички от S биха били приложени автоматично.

Този процес на намиране на това множество T представлява практически интерес.

Тривиални зависимости

Един очевиден начин да се редуцира обема на множеството на ФЗ е да елиминираме тривиалните зависимости.

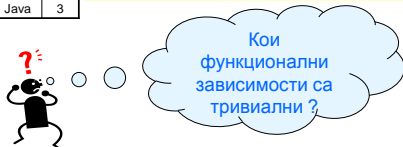
Тривиална зависимост: ФЗ е тривиална, ако дясната страна е подмножество на лявата.

STUDENTS

FNO	YEAR	SUBJ	ASS
021039	Inf-III	DB	4
021039	Inf-III	AI	4
021142	Math-III	DB	5
021142	Math-III	AI	5
021144	Math-III	AI	6
021001	Inf-III	AI	3
021001	Inf-III	SE	3
021001	Inf-III	Java	3

Функционални зависимости:

- $\{FNO\} \rightarrow \{YEAR\}$
- $\{FNO, SUBJ\} \rightarrow \{ASS\}$
- $\{FNO, SUBJ\} \rightarrow \{YEAR\}$
- $\{FNO, SUBJ\} \rightarrow \{YEAR, ASS\}$
- $\{FNO, SUBJ\} \rightarrow \{FNO\}$
- $\{FNO, SUBJ\} \rightarrow \{FNO, SUBJ, YEAR, ASS\}$



STUDENTS

FNO	YEAR	SUBJ	ASS
021039	Inf-III	DB	4
021039	Inf-III	AI	4
021142	Math-III	DB	5
021142	Math-III	AI	5
021144	Math-III	AI	6
021001	Inf-III	AI	3
021001	Inf-III	SE	3
021001	Inf-III	Java	3

Функционални зависимости:

- $\{FNO\} \rightarrow \{YEAR\}$
- $\{FNO, SUBJ\} \rightarrow \{ASS\}$
- $\{FNO, SUBJ\} \rightarrow \{YEAR\}$
- $\{FNO, SUBJ\} \rightarrow \{YEAR, ASS\}$
- ~~$\{FNO, SUBJ\} \rightarrow \{FNO\}$~~
- $\{FNO, SUBJ\} \rightarrow \{FNO, SUBJ, YEAR, ASS\}$



Видове функционални зависимости

• Пълна функционална зависимост

Атрибут е напълно функционално зависим от множество атрибути ако и само ако той е функционално зависим от цялото това множество от атрибути и не е зависим от никое негово подмножество от атрибути.

Пример: нека имаме релацията EXAMS, съдържаща данни за проведените изпити, с ключ (FNO, SUBJ).

FNO	SUBJ	HALL	GRADE

$(FNO, SUBJ) \rightarrow (GRADE)$

Студентският номер и дисциплината определят оценката. Това означава, че студентът може изкара добра оценка по дадена дисциплина, но не е сигурно, че ще изкара добри оценки по всички дисциплини, т.е. оценката зависи и от дисциплината.

• Частична функционална зависимост

Такава е налице, когато само подмножество от съставен детерминант е достатъчно, за да определи функционално даден атрибут.

Тъй като не е правило всички изпити да се провеждат в една и съща зала, още повече за някои от тях има изискване за конкретна зала, то може да се каже, че залата зависи всъщност само от конкретната дисциплина:

(FNO, **SUBJ**) \rightarrow (HALL), т.е.

(SUBJ) \rightarrow (HALL)

• Транзитивна функционална зависимост

Транзитивна ФЗ съществува, когато има „преходна“ функционална зависимост. Т.е., ако $A \rightarrow B$, а $B \rightarrow C$, то $A \rightarrow C$.

LECTURER_ID	FNAME	LNAME	TITLE	SALARY
101	Стоян	Копев	Проф.	800
102	Петър	Иванов	Доц.	600
103	Венета	Георгиева	Ас.	400

В примера LECTURER_ID определя функционално всички атрибути, в частност и TITLE: (LECTURER_ID) \rightarrow (TITLE).

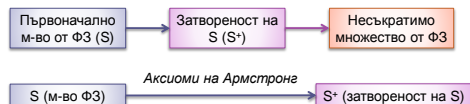
Но званието (длъжността) определя заплатата, така че е налице: (TITLE) \rightarrow (SALARY), т.е. имаме ситуация, в която

(LECTURER_ID) \rightarrow (TITLE)

(TITLE) \rightarrow (SALARY)

Следователно (LECTURER_ID) \rightarrow (SALARY) транзитивно.

Схема на теорията за ФЗ



Аксиомите на Армстронг са правила за извод:

- ✓ Те не ни дават обаче ефективен алгоритъм за изчисляване S^* ;
- ✓ Искаме да намерим ефективен начин за определяне дали една дадена ФЗ е от затвореността.



Важно следствие: при зададено S лесно можем да определим дали една ФЗ $X \rightarrow Y$ може да се изведе от S : **ако и само ако Y е подмножество на затвореността на X^* .**

Аксиоми на Армстронг

Представяват множество от правила за изводи.
Нека A, B, C са произволни подмножества на
множеството от атрибутите на дадена релация R .

Нека AB означава $A \cup B$. Тогава:

- **Рефлексивност:** ако $B \subseteq A$, тогава $A \rightarrow B$
- **Разширяемост:** ако $A \rightarrow B$, тогава $AC \rightarrow BC$
- **Транзитивност:** ако $A \rightarrow B$ и $B \rightarrow C$, тогава $A \rightarrow C$

Всяко от тези правила следва директно от
дефиницията на ФЗ.

От тези правила могат да бъдат изведени други:

- **Самоопределение:** $A \rightarrow A$
- **Декомпозиция:** ако $A \rightarrow BC$, тогава $A \rightarrow B, A \rightarrow C$
- **Обединение:** ако $A \rightarrow B$ и $A \rightarrow C$, тогава $A \rightarrow BC$
- **Композиция:** ако $A \rightarrow B$ и $C \rightarrow D$, тогава $AC \rightarrow BD$
- **Псевдотранзитивност:** ако $X \rightarrow Y$ и $YW \rightarrow Z$, тогава $XW \rightarrow Z$.

Транзитивността е частен случай на псевдотранзитивност,
когато W е null, т.е. псевдотранзитивността е обобщение на
аксиомата за транзитивност.

General unification theorem: ако $A \rightarrow B$ и $C \rightarrow D$,
тогава $A \cup (C - B) \rightarrow D$.

Затвореност на множество от функционални зависимости

Някои функционални зависимости загатват за
други. Например, от

$\{FNO, SUBJ\} \rightarrow \{YEAR, ASS\}$

могат да бъдат изведени следните:

$\{FNO, SUBJ\} \rightarrow YEAR$
 $\{FNO, SUBJ\} \rightarrow ASS$

Затвореност на множество от функционални зависимости

Множеството на всички ФЗ, които следват от едно дадено множество **S** от ФЗ, се нарича затвореност на **S** и се означава с **S⁺**.

Аксиомите на Армстронг са достатъчни, за да бъдат изчислени всички ФЗ от **S⁺**.

Пример:

Нека е дадена релация R с атрибути {A, B, C, D, E, F} и ФЗ {A → BC, B → E, CD → EF}.

Ще покажем, че ФЗ {AD → F} е в сила в R и така тя е член на затвореността на даденото множество.

1. A → BC (дадено)
2. A → C (1, декомпозиция)
3. AD → CD (2, разширение)
4. CD → EF (дадено)
5. AD → EF (3 и 4, транзитивност)
6. AD → F (5, декомпозиция)



Пример:

Нека е дадена релация R с атрибути:

- EMPNO – номер на служител;
- DEPTNO – номер на отдел, в който работи;
- MGRNO – номер на мениджъра му;
- PRJNO – номер на проект, по който работи служителът;
- DEPTNAME – име на отдела;
- PRC – време в %, през което служителът работи по този проект.

EMPNO	DEPTNO	MGRNO	PRJNO	DEPTNAME	PRC
1001	1		20	Изследвания	30%
1002	1	1001	30	Изследвания	40%
1002	1	1001	20	Изследвания	60%

Дадени са следните ФЗ:

- EMPNO → {DEPTNO, MGRNO}
- DEPTNO → DEPTNAME
- {MGRNO, PRJNO} → {DEPTNAME, PRC}

Целта ни е да покажем, че ФЗ {EMPNO, PRJNO} → PRC е в сила за R и така тя е член на затвореността на даденото множество.

- $EMPNO \rightarrow \{DEPTNO, MGRNO\}$
- $DEPTNO \rightarrow DEPTNAME$
- $\{MGRNO, PRJNO\} \rightarrow \{DEPTNAME, PRC\}$

1. $EMPNO \rightarrow \{DEPTNO, MGRNO\}$ (дадено)
2. $EMPNO \rightarrow MGRNO$ (1, декомпозиция)
3. $\{EMPNO, PRJNO\} \rightarrow \{MGRNO, PRJNO\}$ (2, разширение)
4. $\{MGRNO, PRJNO\} \rightarrow \{DEPTNAME, PRC\}$ (дадено)
5. $\{EMPNO, PRJNO\} \rightarrow \{DEPTNAME, PRC\}$ (3 и 4, транзитивност)
6. $\{EMPNO, PRJNO\} \rightarrow PRC$ (5, декомпозиция)



Затвореност на множество от атрибути

Затвореността на едно множество от атрибути е ефективен алгоритъм за определяне принадлежност на една ФЗ в затвореността S^+ . Този алгоритъм използва концепцията за суперключ.

Суперключ: множество от атрибути на една релация, което включва най-малко един КК на релацията като едно подмножество (не е задължително истинско).

Дефиницията на суперключ може следователно да се изведе от тази на КК чрез премахване на изискването за несъкратимост.

Непосредствено следва, че суперключовете за една релация са тези подмножества K на множеството на атрибутите, така че $\Phi Z K \rightarrow A$ важи за всеки атрибут A .

Нека допуснем, че знаем ФЗ, които са в сила за дадена релация и искаме да определим КК-вете за тази релация.

КК по дефиниция са тези суперключове, които са *несъкратими*.

За да определим дали K е суперключ, ние трябва просто да определим дали всички атрибути на релацията функционално зависят от K .

И така, дадено е едно множество S от ФЗ, които са в сила за R , ние се нуждаем от начин за определяне **множеството на всички атрибути на R , които са функционално зависими от K , т.нар. затвореност K^+ на K върху S .**

Сравнително прост алгоритъм за изчисляване на затвореност на множество атрибути

```
CLOSURE[K, S] := K;
do true
  for each FD  $X \rightarrow Y$  in  $S$  do
    if  $X$  е подмножество на CLOSURE[K, S]
      then CLOSURE[K, S] := CLOSURE[K, S] UNION  $Y$ 
  end;

  if CLOSURE[K, S] остава непроменена при тази итерация
    then /*изчислението е завършено*/ exit;
end;
```

Пример: да разгледаме релацията R с атрибути A, B, C, D, E, F и следните ФЗ:

1. $A \rightarrow BC$
2. $E \rightarrow CF$
3. $B \rightarrow E$
4. $CD \rightarrow EF$



Ще изведем затвореността $\{A,B\}^+$ за множеството от атрибути $\{A,B\}$ върху това множество от ФЗ.
Инициализираме $CLOSURE[K, S] = \{A,B\}$;

Вътрешният цикъл повтаряме 4 пъти - по веднъж за всяка от дадените ФЗ:

```
CLOSURE[K, S] := {A, B};
do true
  for each FD X→Y in S do
    if X е подмножество на CLOSURE[K, S]
      then CLOSURE[K, S] := CLOSURE[K, S] UNION Y
  end;

  if CLOSURE[K, S] остава непроменена при тази итерация
    then /*изчислението е завършено*/ exit;
end;
```

1-ва итерация (за $A \rightarrow BC$): - лявата страна е подмножество на $CLOSURE[K, S]$ - добавяме атрибутите B,C към резултата: $CLOSURE[K, S] = \{A,B,C\}$

```
...
// A→BC
if A ⊂ CLOSURE[{A,B}, S]
  then CLOSURE[{A,B}, S] := CLOSURE[{A,B}, S] UNION {B,C}
...
```

2-ра итерация (за $E \rightarrow CF$): - лявата страна не е подмножество на $CLOSURE[K, S]$ - остава непроменена

```
...
// E→CF
if E ⊂ CLOSURE[{A,B,C}, S] - false
...
```

3-та итерация (за $B \rightarrow E$): - лявата страна е подмножество на $CLOSURE[K, S]$ - добавяме атрибутите E към резултата: $CLOSURE[K, S] = \{A,B,C,E\}$

```
...
// B→E
if B ⊂ CLOSURE[{A,B,C}, S]
  then CLOSURE[{A,B,C}, S] := CLOSURE[{A,B,C}, S] UNION {E}
...
```

4-та итерация (за $CD \rightarrow EF$): - лявата страна не е подмножество на $CLOSURE[K, S]$ - остава непроменена

```
...
// CD→EF
if CD ⊂ CLOSURE[{A,B,C,E}, S] - false
...
```

Дотук след първата итерация на външния цикъл имаме $CLOSURE[K, S] = \{A, B, C, E\}$. Обхождаме вътрешния цикъл отново 4 пъти (за всяка функционална зависимост):

1-ва итерация (за $A \rightarrow BC$) - не се променя;

2-ра итерация (за $E \rightarrow CF$) - разширява се, $CLOSURE[K, S] = \{A, B, C, E, F\}$;

```
...
// E → CF
if E ⊆ CLOSURE[{A, B, C, E}, S]
then CLOSURE[{A, B, C, E}, S] := CLOSURE[{A, B, C, E}, S]
    UNION {C, F}
...
```

3-та итерация (за $B \rightarrow E$) - не се променя;

4-та итерация (за $CD \rightarrow EF$) - не се променя.

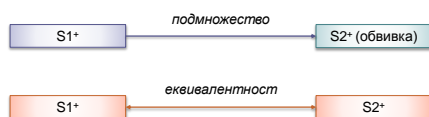
След като има промяна отново обхождаме вътрешния цикъл 4 пъти - $CLOSURE[K, S]$ остава непроменена този път - процесът завършва като $\{A, B\}^+ = \{A, B, C, E, F\} \neq \{A, B, C, D, E, F\}$

Следователно $\{A, B\}$ не е суперключ, следователно не е и кандидат-ключ!

Важно заключение: дадено е едно множество S от ФЗ - можем лесно да определим дали една ФЗ $X \rightarrow Y$ следва от S , защото тази ФЗ ще следва, ако и само ако Y е подмножество на затвореността X^+ на X върху S .

С други думи, сега имаме един прост начин за определяне дали дадена ФЗ $X \rightarrow Y$ е в затвореността на S .

Схема на теорията за ФЗ



Несъкратими множества от зависимости

Обвивка: нека S_1 и S_2 са две множества от ФЗ. Ако всяка ФЗ, изведена от S_1 , се извежда от ФЗ в S_2 - т.е. ако S_1^+ е подмножество на S_2^+ , тогава S_2 е обвивка на S_1 .

Това означава, че ако СУБД приложи ограниченията, представени чрез ФЗ в S_2 , тогава тя ще прилага автоматично и ФЗ в S_1 .

Еквивалентност: множествата от ФЗ S_2 и S_1 са еквивалентни, ако S_2 е обвивка на S_1 и S_1 е обвивка на S_2 .

Ако S_1 и S_2 са еквивалентни, тогава когато СУБД прилага ограниченията, представени чрез ФЗ в S_1 , тя ще прилага автоматично и ФЗ в S_2 и обратно.

Несъкратимо множество от ФЗ: ако и само ако то удовлетворява следните три свойства:

- дясната страна на всяка ФЗ в S включва точно един атрибут;
- лявата страна на всяка ФЗ в S е несъкратима по ред - означава, че никой атрибут не може да бъде премахнат от нея без да промени затвореността ѝ (т.е. без конвертиране на S в някакво множество нееквивалентно на S), така ФЗ е *ляво-несъкратима*;
- никоя ФЗ в S не може да бъде отстранена от S без да промени затвореността ѝ - т.е. без да трансформира S в някакво множество нееквивалентно на S.

Пример

Нека разгледаме релацията P, съдържаща данни за частите. Следните ФЗ са в сила:

- $P_NO \rightarrow PNAME$
- $P_NO \rightarrow COLOR$
- $P_NO \rightarrow WEIGHT$
- $P_NO \rightarrow CITY$

Лесно се вижда, че това множество е несъкратимо:

- ✓ дясната част е от един атрибут на всяка ФЗ;
- ✓ лявата също е несъкратима;
- ✓ никоя ФЗ не може да бъде премахната без промяна на затвореността (т.е. *без загуба на информация*).

Пример

От друга страна, следните множества от ФЗ не са несъкратими:

1. $P_NO \rightarrow \{PNAME, COLOR\}$ – дясната част не е единичен атрибут.

$P_NO \rightarrow WEIGHT$
 $P_NO \rightarrow CITY$

2. $\{P_NO, PNAME\} \rightarrow COLOR$ – PNAME може да бъде премахнат от лявата част без промяна на затвореността, т.е. тази зависимост не е ляво-несъкратима.

$P_NO \rightarrow PNAME$
 $P_NO \rightarrow WEIGHT$
 $P_NO \rightarrow CITY$

3. $P_NO \rightarrow P_NO$ – тази ФЗ може да бъде премахната без промяна на затвореността.

$P_NO \rightarrow PNAME$
 $P_NO \rightarrow WEIGHT$
 $P_NO \rightarrow COLOR$
 $P_NO \rightarrow CITY$

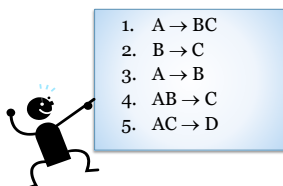
Извод

Вече можем да твърдим, че за всяко множество от ФЗ съществува поне едно еквивалентно на него, което е несъкратимо.

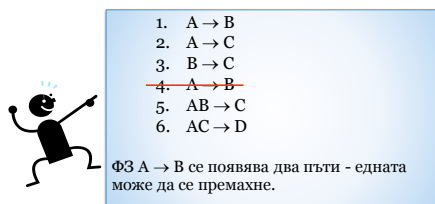
Напр. нека имаме множество от ФЗ S.

- Благодарение на правилото за декомпозиция можем да допуснем, че всяка ФЗ от S има единична дясна страна;
- След това за всяка ФЗ f от S преглеждаме всеки атрибут A от лявата страна; ако S е еквивалентно на множеството от ФЗ, получено чрез премахване на A от лявата страна на f, то премахваме A от лявата страна на f;
- За всяка ФЗ f от S проверяваме ако S и (S-f) са еквивалентни, премахваме f от S.

Пример: да разгледаме релацията R с атрибути A,B,C,D. Следните ФЗ са в сила:



1. Декомпозираме ФЗ така, че всички да останат с единична дясна част.



2. Атрибутът С може да се елиминира от лявата страна на $AC \rightarrow D$: от $A \rightarrow C$ следва $A \rightarrow AC$ (чрез разширение) и е дадена $AC \rightarrow D$.

Така $A \rightarrow D$ (чрез транзитивност) - следователно С в лявата страна на $AC \rightarrow D$ е излишно.

1. $A \rightarrow B$
2. $A \rightarrow C$
3. $B \rightarrow C$
4. $AB \rightarrow C$
5. $AC \rightarrow D$ - става $A \rightarrow D$
6. ~~$A \rightarrow D$~~ - повтаря се вече



3. $AB \rightarrow C$ може да бъде елиминирана – от $A \rightarrow C$ следва $AB \rightarrow CB$ (чрез разширение) и така $AB \rightarrow C$ (чрез декомпозиция).

1. $A \rightarrow B$
2. $A \rightarrow C$
3. $B \rightarrow C$
4. ~~$AB \rightarrow C$~~
5. $A \rightarrow D$



4. $A \rightarrow C$ е изведена чрез ФЗ $A \rightarrow B$ и $B \rightarrow C$, така че тя също може да се елиминира - така остават:

1. $A \rightarrow B$
2. ~~$A \rightarrow C$~~
3. $B \rightarrow C$
4. $A \rightarrow D$

Това множество е несъкратимо!



Типове зависимости - обобщение

• Функционални зависимости

Функционална зависимост (ФЗ) съществува между две полета **A** и **B**, когато единична стойност на **A** е директно свързана с единична стойност на **B**. Това означава, че по дадена стойност на **A** за определен запис в таблица винаги може да бъде извлечена стойността на **B** в този запис.

Функционалната зависимост се бележи така: **A→B**, като елементът отляво на стрелката се нарича *детерминант*, а този отдясно - *зависим*.

Казваме, че стойността на **A** определя стойността на **B** или стойността на **B** е функционално зависима от стойността на **A**.

Функционални зависимости

Коректно проектираните таблици винаги съдържат добре дефинирани функционални зависимости. За пример може да послужи първичният ключ, който функционално определя всички неключови полета в таблицата – по зададена стойност на първичния ключ за определен запис можем да извлечем стойностите на всички неключови полета от запис. В посочената таблица полето CustomerId определя всички останали полета в таблицата.

CustomerId	FirstName	LastName	City	Address	ZIP	AreaCode
1001	Иван	Петров	Пловдив	Бул. България - 180	4000	032
1002	Надежда	Георгиева	София	Дружба 2, бл. 315	1000	02
1003	Милен	Иванов	Петрич	Ул. Победа – 12	4510	0324
1004	Николай	Стефанов	Ямбол	Ул. Диана - 23	4300	045

• Транзитивни зависимости

Нека имаме три полета **A**, **B** и **C**, между които съществуват следните функционални зависимости: **A→B**, **B→C**.

Транзитивна зависимост (ТЗ) съществува между **A** и **C**, защото конкретна стойност на **A** е индиректно асоциирана с конкретна стойност на **C** посредством **B**. Логиката е следната:

A определя стойността на **B**

B определя стойността на **C**

Следователно **A** транзитивно определя стойността на **C**.

Транзитивната зависимост се бележи така: **A⇒C**

Казваме, че стойността на **A** транзитивно определя стойността на **C** (чрез **B**) или стойността на **C** е транзитивно зависима от стойността на **A** (чрез **B**).

Транзитивни зависимости

Една добре проектирана таблица представя един и само един обект. Обаче таблица, която съдържа транзитивни зависимости, ще представя два или повече обекта, в зависимост от броя на транзитивните зависимости в нея – например таблица с една транзитивна зависимост ще представя два обекта, с две – три обекта и т.н. Такава таблица е некоректно проектирана и ще представлява субект на аномалии на промените.

Следната таблица има транзитивна зависимост между полето EmpID и полетата за отдела. Логиката е следната:

- EmpID определя DeptID
- DeptID определя DeptName

По този начин EmpID транзитивно определя DeptName.

EmpID	FirstName	LastName	Phone	DeptID	DeptName
101	Петър	Станев	76-85-43	1	Администрация
102	Ивайло	Колев	43-87-12	3	Счетоводство
103	Станимира	Колев	44-54-10	3	Счетоводство
104	Никола	Петров	82-56-90	2	Човешки ресурси

• Мулти-стойностни (multi-valued) зависимости

Мулти-стойностна зависимост (МСЗ) е налице между две полета **A** и **B**, когато единична (конкретна) стойност на **A** е директно асоциирана с две или повече стойности на **B**.

Мулти-стойностна зависимост се бележи така:

A →→ B

Казваме, че конкретна стойност на **A** определя множество стойности на **B** или множество стойности на **B** са функционално зависими от стойност на **A**.

Мулти-стойностна зависимост може да съществува на ниво поле или ниво запис, и две или повече независими МСЗ могат да съществуват в една таблица едновременно.

Мулти-стойностни зависимости

МСЗ са подобни на транзитивните зависимости в смисъл, че тяхното наличие означава, че таблицата представя два или повече обекта. Такава таблица, естествено, е некоректно проектирана и е субект на аномалиите на промените. Представената таблица съдържа МСЗ на ниво поле. Тази таблица съдържа служители и комисии, в които те участват. МСЗ е налице между полетата EmpID и Committees – една стойност на EmpID е асоциирана с една или много стойности на Committees. Въпреки че не е съвсем очевидно, тази таблица представя поне два обекта – служители и комисии, съставени от служители.

EmpID	FirstName	LastName	Phone	Committees
101	Петър	Станев	76-85-43	Управление
102	Ивайло	Колев	43-87-12	Сигурност, Инвестиции
103	Станимира	Василева	44-54-10	Инвестиции, Управление, Сигурност
104	Никола	Петров	82-56-90	Управление, Сигурност
105	Димитър	Иванов	77-56-90	

Мулти-стойностни зависимости

Следващата таблица също съдържа МСЗ, но на ниво запис. Зависимостта е отново между полетата EmpID и Committees. В този случай, обаче, данните за служителя се повтарят за всяка комисия, в която той участва. Тази таблица отново представя двата обекта – служители и комисии, съставени от служители.

EmpID	FirstName	LastName	Phone	Committees
101	Петър	Станев	76-85-43	Управление
102	Ивайло	Колев	43-87-12	Сигурност
102	Ивайло	Колев	43-87-12	Инвестиции
103	Станимира	Василева	44-54-10	Инвестиции
103	Станимира	Василева	44-54-10	Управление
103	Станимира	Василева	44-54-10	Сигурност
104	Никола	Петров	82-56-90	Управление
104	Никола	Петров	82-56-90	Сигурност
105	Димитър	Иванов	77-56-90	

Мулти-стойностни зависимости

Пример за две независими МСЗ – едната е между полетата Employee и Language, другата е между Employee и Certificate. Налице е повторение на данните за всеки служител, когато трябва да се регистрира факта, че той владее език или има сертификат. Тези две зависимости казваме, че са независими една от друга, защото владенето на английски не е задължително условие, за да бъде служителят разработчик на системи с MS SQL Server 2000. От друга страна сертификатът за Visual Basic 6 не е задължително условие за владенето на немски език.

Employee	Language	Certificate
Петър Станев	Английски	
Ивайло Колев	Немски	
Ивайло Колев	Руски	
Ивайло Колев		MS SQL Server 2000
Станимира Василева	Английски	
Станимира Василева		Visual Basic 6
Станимира Василева		C++ Builder 5
Никола Петров	Френски	
Никола Петров		MS SQL Server 2000
Димитър Иванов		

• JOIN зависимости

JOIN зависимост (JD) съществува в една таблица **A**, ако всеки запис в таблицата може да бъде реконструиран от SQL JOIN операция, която съединява отново всички таблици, създадени при нейната декомпозиция. Това твърдение трябва да е истина за всички съществуващи записи в **A** към момента на нейната декомпозиция и за всеки валиден запис, който би могъл да бъде въведен преди декомпозицията ѝ. Записи, добавени в декомпозираните таблици, трябва да могат да формират валиден запис за **A**, когато те бъдат обединени чрез JOIN. Освен това, никакви записи не трябва да бъдат изгубени и никакви фалшиви да бъдат добавени при тази декомпозиция.

JOIN зависимости

Следващата таблица съдържа данни за продавачи. Можем да кажем, че таблицата има JOIN зависимост, защото тя може да бъде декомпозирана на по-малки таблици (това не означава, че задължително трябва да бъде декомпозирана) – една за статусите и една за данните на продавачите.

ID	Name	City	Phone	Status	Discount
1	ЕТ "Иван Иванов"	София	943-12-34	Предпочитан	10
2	"Балкан" ООД	Смолян	3-45-12	Предпочитан	10
3	"Орион" ООД	Пазарджик	43-88-90	При необходимост	5
4	ЕТ "Илия Колев"	Пловдив	95-67-44	Обикновен	7
5	ЕТ "Лилия-2000"	София	943-44-54	Предпочитан	10

Status	Discount
Предпочитан	10
При необходимост	5
Обикновен	7

ID	Name	City	Phone	Status
1	ЕТ "Иван Иванов"	София	974-34-23	Предпочитан
2	"Балкан" ООД	Смолян	3-45-12	Предпочитан
3	"Орион" ООД	Пазарджик	43-88-90	При необходимост
4	ЕТ "Илия Колев"	Пловдив	95-67-44	Обикновен
5	ЕТ "Лилия-2000"	София	943-44-54	Предпочитан

JOIN зависимости

Поради наличието на JD трябва да можем да изпълним долната заявка, която да създаде наново оригиналната таблица, като при този процес не трябва да има загуба на данни, нито поява на фалшиви записи.

```
SELECT Trader.ID, Trader.Name, Trader.City,
       Trader.Phone, Status.Discount, Status.Status
FROM Trader NATURAL JOIN Status
```

Няма изискване всяка таблица да притежава JD. Всъщност таблиците, които са кандидати за JD са тези, които могат да бъдат декомпозирани на по-малки таблици. Веднъж след като декомпозицията е извършена, споменатите правила трябва да важат, за да бъде JD валидна.