SLOTS 04

CHAPTER 3 (CONT.)

DESIGN THEORY FOR RELATIONAL DATABASES

II.

DECOMPOSITION

2.1. DECOMPOSITION

- The accepted way to eliminate anomalies is the *decomposition* of relations
- Decomposition of a relation R involves splitting the attributes of R to make the schemas of 2 new relations

2.1. DECOMPOSITION

Definition:

Given a relation $R(A_1,\ldots,A_n)$, we say R is decomposed into $S(B_1,\ldots,B_m)$ and $T(C_1,\ldots,C_k)$ if:

+ $\{A_1,\ldots,A_n\} = \{B_1,\ldots,B_m\} \cup \{C_1,\ldots,C_k\}$

+ $S = \prod_{B1,\ldots Bm}(R)$

+ $T = \prod_{C1,\ldots,Ck}(R)$

Example: Decomposition

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With The Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

title	*year*	*length*	*genre*	*studioName*
Star Wars	1977	124	SciFi	Fox
Gone With The Wind	1939	231	drama	MGM
Wayne's World	1992	95	comedy	Paramount

title	*year*	*starName*
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Gone With	1939	Vivien Leigh
Wayne's W	1992	Dana Carvey
Wayne's W	1992	Mike Meyers

Discuss

title	*year*	*length*	*genre*	*studioName*
Star Wars	1977	124	SciFi	Fox
Gone With The Wind	1939	231	drama	MGM
Wayne's World	1992	95	comedy	Paramount

title	*year*	*starName*
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Gone With	1939	Vivien Leigh
Wayne's W	1992	Dana Carvey
Wayne's W	1992	Mike Meyers

- The redundancy is eliminated (the length of each film appears only once)
- The risk of an update anomaly is gone (we only have to change the length of *Star Wars* in one tuple)
- The risk of a deletion anomaly is gone (if we delete all the stars for *Gone with the wind,* that deletion makes the movie disappear from the right but still be found in the left)

2.1. DECOMPOSITION

The Good & Bad

- We observed that before we decompose a relation schema into BCNF, it can exhibit anomalies; That's the "Good"
- However, decomposition can also have some bad:
  - Maybe we can't recovery the original information; OR
  - After reconstruction, the FDs maybe not hold

Example: Loss of information after decomposition

R

A	B	C
1	2	3
4	2	5

R1

A	B
1	2
4	2

R2

B	C
2	3
2	5

R3

A	B	C
1	2	3
1	2	5
4	2	3
4	2	5

- Suppose we have R(A,B,C) but neither of the FD's B->A nor B->C holds.
- R is decomposed into R1 and R2 as above
- When we try to re-construct R by Natural Join of R1 and R2, we have: R3 = R1 X R2 (but R3 <> R1 => We lost information)

Example: Dependency Loss

- If we check the projected FD's in the relations of the decomposition, can we can be sure that when we reconstruct the original relation from the decomposition by joining, the result will satisfy the original FD's?

NORMALIZATION

III.1. DEFINITIONS

- ❑ ***Multivalued Attributes*** *(thuộc tính đa trị)*
- ❑ ***Atomic values*** *(thuộc tính nguyên tố)*
- ❑ ***non-key attribute*** *(thuộc tính không khoá)*
- ❑ ***Partial Dependency*** *(phụ thuộc bộ phận)*
- ❑ ***Transitive Dependency*** *(phụ thuộc bắc cầu)*

1. Multivalued Attributes

- ***Multivalued Attributes*** (or ***repeating groups***): non-key attributes or groups of non-key attributes the values of which are not uniquely identified by (directly or indirectly) (not functionally dependent on) the value of the Primary Key (or its part).

Multivalued Attributes

Multi Value
Or repeating
groups

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	\$50 \$50 \$100	B A B+

FDs = { StudentID→StudentName, Address, HouseName, HouseColor,
Subject→SubjectCost,
StudentID, subject →Grade}

2. *Partial Dependency*

- ***Partial Dependency*** – when a non-key attribute is determined by a part, but not the whole, of a **COMPOSITE** primary key.

3. *Transitive Dependency*

- ***Transitive Dependency*** – when a non-key attribute determines another non-key attribute.

III.2. NORMAL FORMS

- First Normal Form
- Second Normal Form
- Third Normal Form
- Boyce-Codd Normal Form
- Fourth Normal Form
- Fifth Normal Form
- Domain-Key Normal Form

First normal form 1NF

- 1NF A relation R is in first normal form (1NF)

if and only if

all underlying domains contain atomic values only.

Is it 1NF?

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	$50 $50 $100	B A B+

1NF

No 1NF. There are repeating groups (subject, subjectcost, grade)

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	$50 $50 $100	B A B+

How can you make it 1NF?

1NF

Create new rows so each cell contains only one value

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	$50 $50 $100	B A B+

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

But now look – is the *studentID* primary key still valid?

1 NF

No – the studentID no longer uniquely identifies each row

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

You now need to declare *studentID* and *subject* together to uniquely identify each row.

So the new key is **StudentID *and* Subject**.

1NF

So. We now have 1NF.

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

Is it 2NF?

Second normal form 2NF

A relation R is in **2NF**

if and only if

it is in 1NF and every non-key attribute is fully dependent on the primary key

2NF

StudentName & **Address** are dependent on **studentID** (which is part of the key)

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

But they are not dependent on ***Subject*** (the *other* part of the key)

2NF

And 2NF requires…

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

All non-key fields are dependent on the ENTIRE key (studentID + subject)

So it's not 2NF

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

How can we fix it?

Convert to
2NF

2NF

Make new tables

- Make a new table for each primary key field
- Give each new table its own primary key
- Move columns from the original table to the new table that matches their primary key…

CONVERT TO 2NF

Step 1

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

CONVERT TO 2NF

Step 2

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

CONVERT TO 2NF

Step 3

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

RESULTS TABLE (key = StudentID+Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

CONVERT TO 2NF

Step 4 - relationships

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

RESULTS TABLE (key = StudentID+Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

Each student can only appear ONCE in the **student** table

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

RESULTS TABLE (key = StudentID+Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

Each **subject** can only appear ONCE in the **subjects** table

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

RESULTS TABLE (key = StudentID+Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

A **subject** can be listed **MANY times** in the **results** table (for different students)

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

RESULTS TABLE (key = StudentID+Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

A **student** can be listed **MANY times** in the **results** table (for different subjects)

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

RESULTS TABLE (key = StudentID+Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

SubjectCost is only dependent on the primary key, ***Subject***

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseNam
19594332X	Mary Watson	10 Charles Street	Bob

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

RESULTS TABLE (key = StudentID+Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

2NF CHECK

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

1

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

1

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

Grade is only dependent on the primary key (*studentID + subject*)

2NF CHECK

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

Name, Address are only dependent on the primary key (*StudentID*)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

So it is 2NF!

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

1

SUBJECTS TABLE (key = Subject)

1

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

∞

∞

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

But is it 3NF?

RESULTS TABLE (key = StudentID+Subject)

Third normal form 3NF

A relation R is in **3NF**

if and only if

it is **in 2NF and every non-key attribute is non-transitively dependent on the primary key**

Third normal form 3NF

An attribute **C** is **transitively dependent** on attribute **A** if there exists an attribute B such that:

A->B and **B->C**

3NF

Note that

- 3NF is concerned with **transitive dependencies** (which do not involve candidate keys).
- A relation with more than one candidate key will clearly have transitive dependencies of the form:

primary_key -> other_candidate_key -> any_non-key_column

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

RESULTS TABLE (key = StudentID+Subject)

3NF CHECK

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

StudentID-> HouseName

and

HouseName-> ***HouseColour***

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

3NF CHECK

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

But either way, non-key fields are dependent on MORE THAN THE PRIMARY KEY (studentID)

1

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

1

∞ ∞

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

3NF CHECK

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

And 3NF says that non-key fields must depend on nothing but the key

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

3NF CHECK

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

1

WHAT DO WE DO?

SUBJECTS TABLE (key = Subject)

1

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

∞ ∞

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

Again, carve off the offending fields

StudentTable

StudentID	StudentName	Address	HouseName
19594332X	Mary Watson	10 Charles Street	Bob

Primary key: StudentID

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

3NF FIX

StudentTable

StudentID	StudentName	Address	HouseName
19594332X	Mary Watson	10 Charles Street	Bob

Primary key: StudentID

HouseTable

HouseName	HouseColor
Bob	Red

Primary key: HouseName

1

∞

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

∞

RESULTS TABLE (key = StudentID+Subject)

1

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

3NF FIX

StudentTable

StudentID	StudentName	Address	HouseName
19594332X	Mary Watson	10 Charles Street	Bob

Primary key: StudentID

∞

1

HouseTable

HouseName	HouseColor
Bob	Red

Primary key: HouseName

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

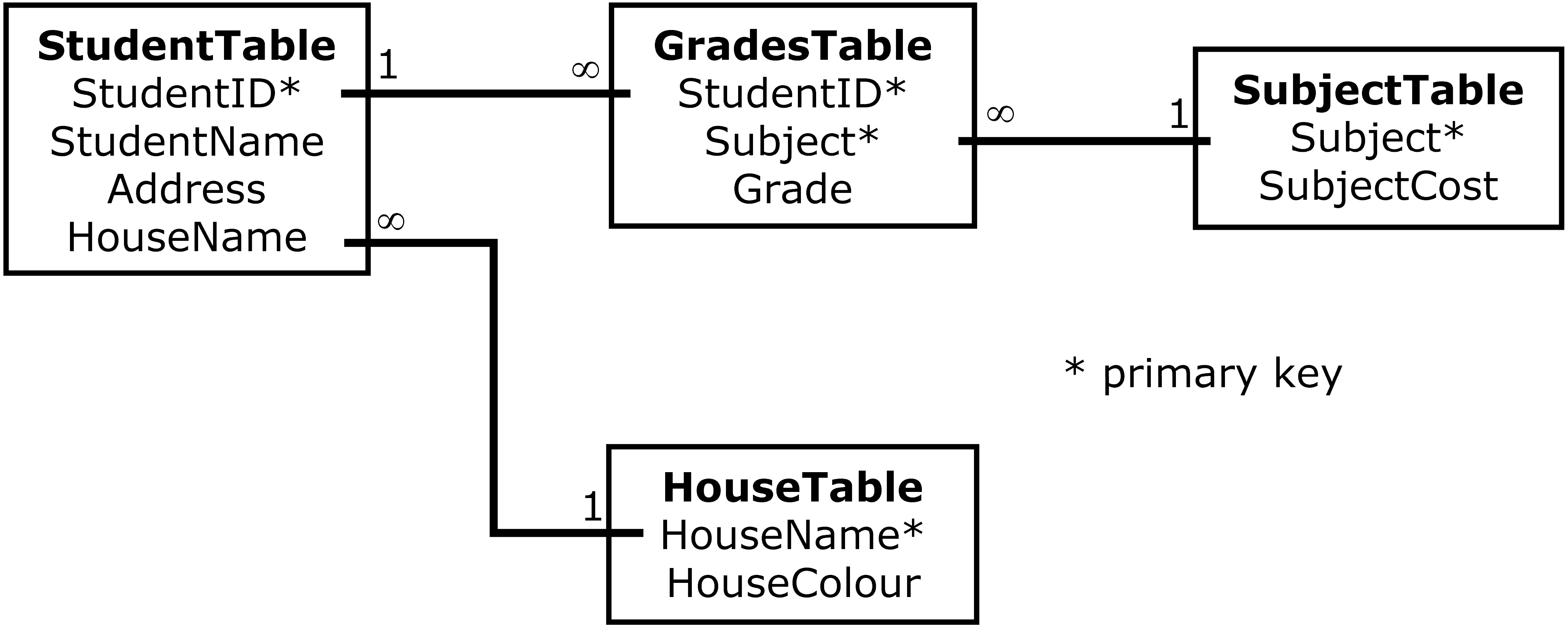
∞ ∞ 1

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

3NF FIX

Or…


StudentTable
StudentID*
StudentName
Address
HouseName
GradesTable
StudentID*
Subject*
Grade
SubjectTable
Subject*
SubjectCost
HouseTable
HouseName*
HouseColour
1
∞
∞
1
∞
1
* primary key

3NF FIX

StudentTable

StudentID	StudentName	Address	HouseName
19594332X	Mary Watson	10 Charles Street	Bob

Primary key: StudentID

HouseTable

HouseName	HouseColor
Bob	Red

Primary key: HouseName

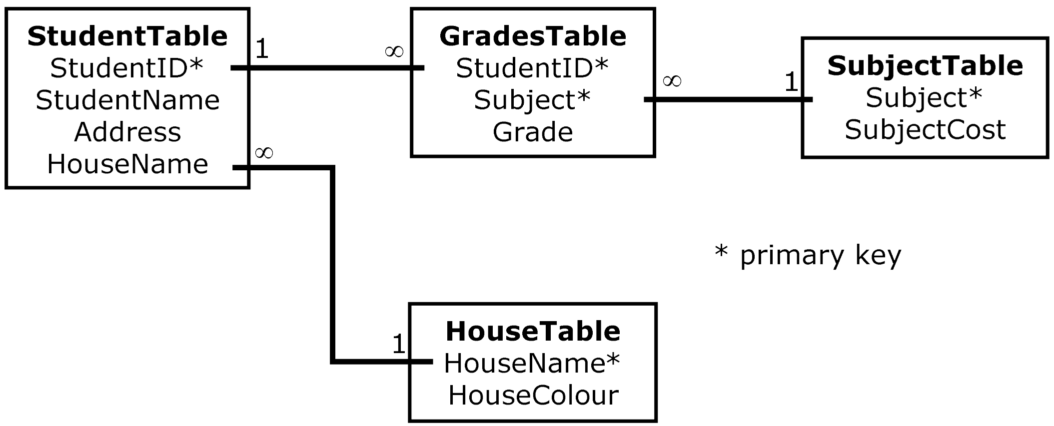
StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

SUBJECTS TABLE (key = Subject)

Or…



* primary key

The Reveal

Before...

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	$50 $50 $100	B A B+

After...

StudentTable

StudentID	StudentName	Address	HouseName
19594332X	Mary Watson	10 Charles Street	Bob

Primary key: StudentID

∞ 1

HouseTable

HouseName	HouseColor
Bob	Red

Primary key: HouseName

1 ∞

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

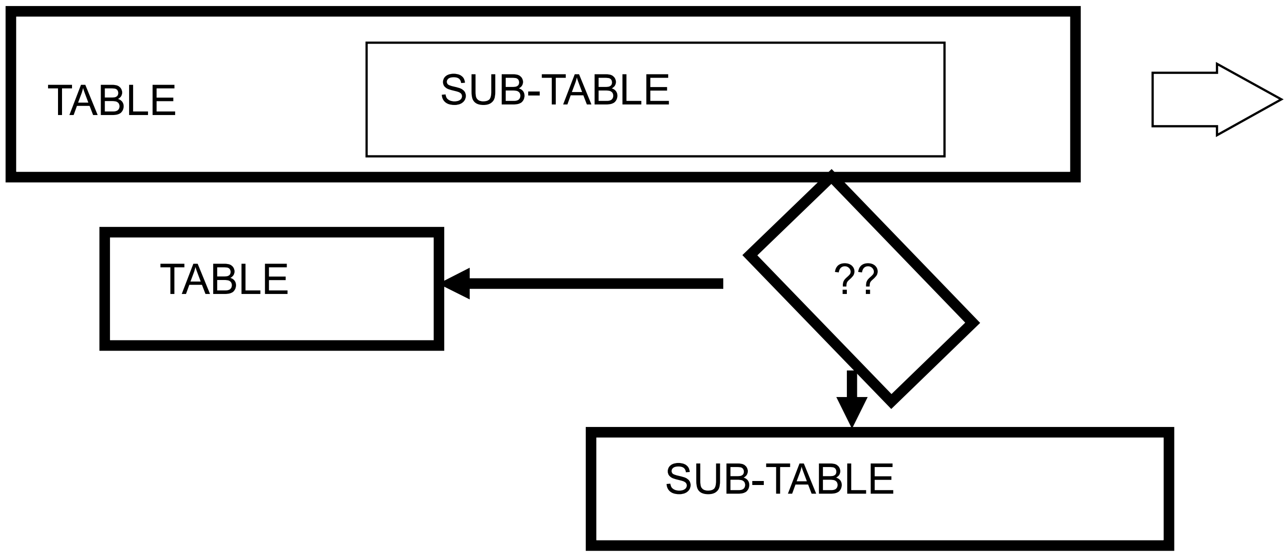
∞ 1

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

SUBJECTS TABLE (key = Subject)

3NF
No transitive dependencies

Table contains data from an embedded entity with non-key attributes.



BCNF is the same, but the embedded table may involve key attributes.

BCNF

- A relation R is in BCNF if and only if:
  Whenever there is a Non-Trivial FD $A_1A_2..A_n$ -> $B_1B_2..B_m$ for R, it is the case that:
  $\{A_1,...,A_n\}$ is a super-key for R

- That is: the left side of every Non-Trivial FD must be a super-key

BCNF decomposition algorithm (self studying)

- **Input**: A relation R with a set of FD's F
- **Output**: A BCNF decomposition of R with lossless join
- **Method**:
  - At each step compute the key for the sub-relation R
  - if not in BCNF, pick any FD X->Y which violates
  - break the relation into 2 sub-relations
    - R1(XY)
    - R2(S - Y)
    - this has a lossless join
    - project FD's onto each sub-relation
  - continue until no more offending FD's

3NF decomposition algorithm – self studying

- **Input**: A relation R with a set of FD's F
- **Output**: A decomposition of R into a collection of relations, all of which are in 3NF. This decomposition has a lossless join and dependency-preservation.
- **Method**:
  - Find minimal basic for F, say G.
  - $\forall$ X-A $\in$ G, use XA as the schema of one relations in the decomposition.
  - If none of the sets of relations from Step 2 is a super key for R, add another relation whose schema is a key for R.

Summary 1

- Decompose a relation into BCNF is a solution for eliminating anomalies
- But BCNF can cause information loss and dependency loss
- 3NF is a relax solution of BCNF that keep loss-less join and dependency-preservation properties

Summary 2:

2NF	3NF	Boyce-Codd
every nonprime attribute *A* in *R* is not partially dependent on *any* key of R	a *nontrivial* functional dependency: *X* => *A* holds in *R,* either *(a)* *X* is a superkey of *R,* or (b) (b) *A* is a prime attribute of *R.*	a *nontrivial* functional dependency *X* => *A* holds in *R,* then: *a)* *X* is a superkey of *R*

Note:A functional dependency *X* => *Y* is a **full functional dependency** if removal of any attribute *A* from *X* means that the dependency does not hold any more; A **partial functional dependency** is not a **full functional dependency**

EXERCISE 1

Exercise 3.5.2: Consider the relation `Courses`(C, T, H, R, S, G), whose attributes may be thought of informally as course, teacher, hour, room, student, and grade. Let the set of FD's for `Courses` be $C \rightarrow T$, $HR \rightarrow C$, $HT \rightarrow R$, $HS \rightarrow R$, and $CS \rightarrow G$. Intuitively, the first says that a course has a unique teacher, and the second says that only one course can meet in a given room at a given hour. The third says that a teacher can be in only one room at a given hour, and the fourth says the same about students. The last says that students get only one grade in a course.

a) What are all the keys for `Courses`?

b) Verify that the given FD's are their own minimal basis.

c) Use the 3NF synthesis algorithm to find a lossless-join, dependency-preserving decomposition of R into 3NF relations. Are any of the relations not in BCNF?

EXERCISE 2

Exercise 3.5.3: Consider a relation `Stocks`(B, O, I, S, Q, D), whose attributes may be thought of informally as broker, office (of the broker), investor, stock, quantity (of the stock owned by the investor), and dividend (of the stock). Let the set of FD's for `Stocks` be $S \rightarrow D$, $I \rightarrow B$, $IS \rightarrow Q$, and $B \rightarrow O$. Repeat Exercise 3.5.2 for the relation `Stocks`.