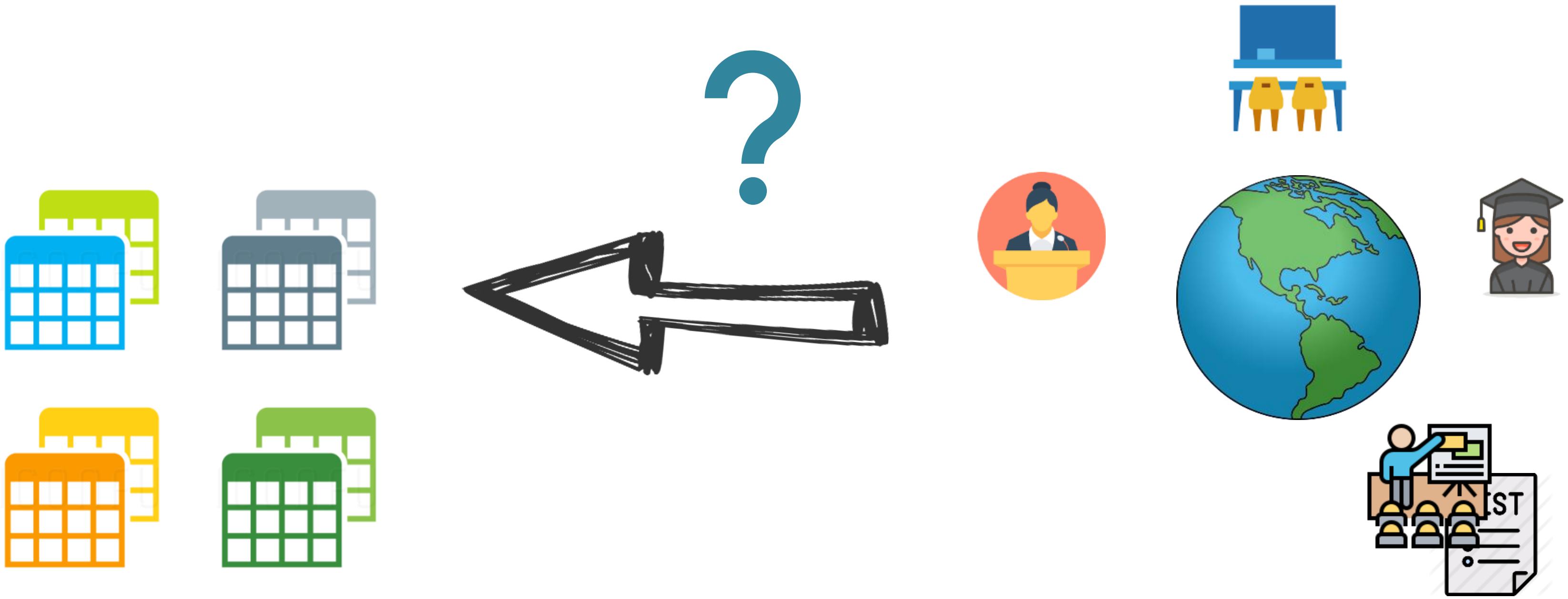
Normalization

Amos Azaria, Netanel Chkroun

Designing a Database



Normalization

- *Database normalization* is the process of structuring a database, usually a *relational database*, in accordance with a series of so-called *normal forms* in order to reduce data *redundancy* and improve data *integrity*.
- It was first proposed by *Edgar F. Codd* as part of his relational model.

https://en.wikipedia.org/wiki/Database_normalization

Dependencies

- An attribute (or set of attributes), B, is said to be dependent of another attribute (or set of attributes), A, if there exists a relation (function) such that A → B.
- In other words, if given A, it is not possible for an entry to have two different values for B, we say that A→B.
- For example, A = student ID, B=student first name.

    student ID -> student first name

- This dependency is also called functional dependency (B is functionally dependent of A).

Dependencies

- Obviously, for every B such that $B \subseteq A$, we have that A → B.
  - E.g.: A = stFirstName, stLastName. B = stFirstName
  - F(stFirstName, stLastName) = stFirstName

A	B	Dependency?
{Street, HouseNum, City, State}	Zip code	A→B
Day of week	Date = {Day, Month, Year}	B→A
First Name	Last Name	None
{University, Department}	DepartmentHeadId	A→B and B→A

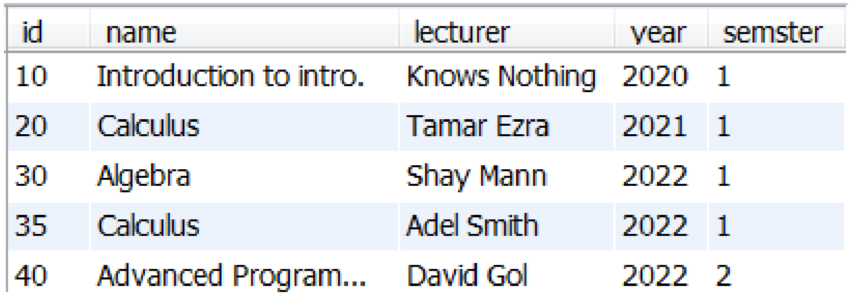
Keys

- Candidate key: A minimal set of attributes that determines an entry. That is, all other attributes are dependent on the key.

Minimal set: removal of any attribute from the set, will no longer determine the entry.

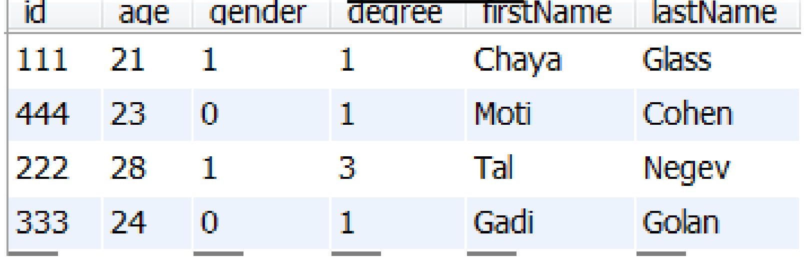
- E.g.:
  - – Student id in student table.
  - – Course table: {id} or {name, year, semester}.

Courses



id	name	lecturer	year	semster
10	Introduction to intro.	Knows Nothing	2020	1
20	Calculus	Tamar Ezra	2021	1
30	Algebra	Shay Mann	2022	1
35	Calculus	Adel Smith	2022	1
40	Advanced Program...	David Gol	2022	2

Students

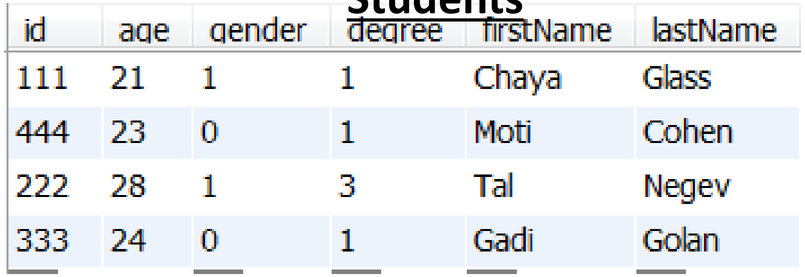


id	age	gender	degree	firstName	lastName
111	21	1	1	Chaya	Glass
444	23	0	1	Moti	Cohen
222	28	1	3	Tal	Negev
333	24	0	1	Gadi	Golan

Keys (cont.)

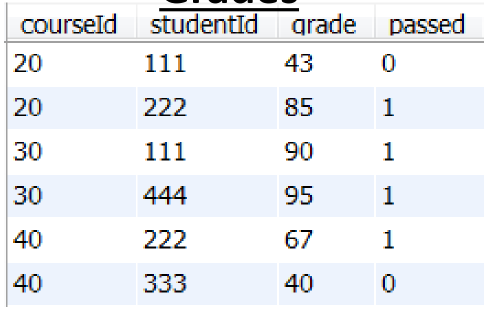
- Is studentName a key?
  - No (there may be multiple students with the same name)
- What would be a key for the grades table?
  - StudentId + courseId

Students



id	age	gender	degree	firstName	lastName
111	21	1	1	Chaya	Glass
444	23	0	1	Moti	Cohen
222	28	1	3	Tal	Negev
333	24	0	1	Gadi	Golan

Grades



courseId	studentId	grade	passed
20	111	43	0
20	222	85	1
30	111	90	1
30	444	95	1
40	222	67	1
40	333	40	0

Keys (cont.)

- A single table can have more than one set of keys (both being minimal), e.g.:
  - R(university, department, depHeadId)
    - {depHeadId}
    - {university, department}

Assuming every department has a single head, and a person can be a department head of a single department in a single university.

Prime / Non-Prime

- Prime attributes are attributes that are part of some candidate-key.
- Similarly, non-prime attributes are attributes that are not part of any candidate-key.

Prine / Non Prime (cont.)

- E.g.:

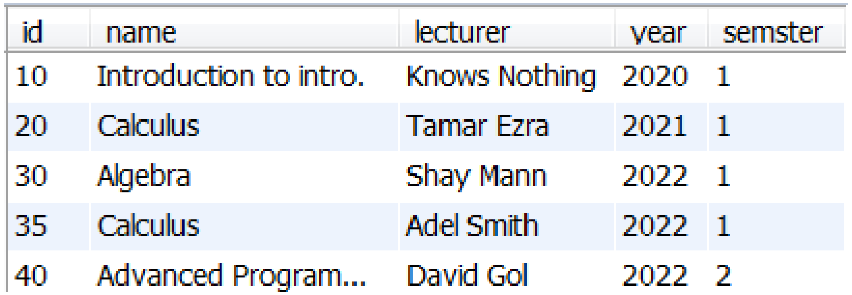
R(university, department, depHeadId)

- Candidate keys are:
  - {depHeadId}
  - {university, department}
- Prime ? Non prime?

Course table:

- Candidate keys are:
  - {id}
  - {name, year, semester}
- Prime? Non prime?

Courses



id	name	lecturer	year	semster
10	Introduction to intro.	Knows Nothing	2020	1
20	Calculus	Tamar Ezra	2021	1
30	Algebra	Shay Mann	2022	1
35	Calculus	Adel Smith	2022	1
40	Advanced Program...	David Gol	2022	2

Super-Key

- **Any** set of attributes that determines an entry.
  - E.g. the whole set of attributes.
- Same as candidate key, just without the minimal requirement.

Candidate kay is called also- ‘Minimal super key’

Normalization

- What is the problem with the following relation?

StudentId	StudentFirst	StudentLast	Courses
542	Yossi	Agasi	4244, 3423, 6734
956	Tamar	Atiya	4244, 5437
754	Gabbi	Matar	4325, 6543, 564
327	Shay	Shalom	5324

Multiple values for a single attribute. How can we get all students in 3423?

Normalization

Heavy redundancy.
What happens when we update student's address? And what if we delete all grades of a student?

- And with this one?

StudentId	StudentFirst	StudentLast	Address	CourseId	Grade
542	Yossi	Agasi	Harambam 45, Ariel	4244	87
542	Yossi	Agasi	Harambam 45, Ariel	3423	65
956	Tamar	Atiya	Hadekel 12, Herzeliya	4244	86
542	Yossi	Agasi	Harambam 45, Ariel	6734	80

Normalization

- *Database normalization* is the process of structuring a database, usually a relational database, in accordance with a series of so-called *normal forms* in order to reduce data redundancy and improve data integrity.
- It was first proposed by *Edgar F. Codd* as part of his relational model.

https://en.wikipedia.org/wiki/Database_normalization

1NF (=Normalized Form)

- Every attribute must hold a single atomic value (searchability)

StudentId	StudentFirst	StudentLast	Courses
542	Yossi	Agasi	4244, 3423, 6734
956	Tamar	Atiya	4244, 5437
754	Gabbi	Matar	4325, 6543, 564
327	Shay	Shalom	5324

StudentId	StudentFirst	StudentLast	Courses
542	Yossi	Agasi	4244
956	Tamar	Atiya	4244
754	Gabbi	Matar	4325
327	Shay	Shalom	5324
542	Yossi	Agasi	3423
542	Yossi	Agasi	6734
956	Tamar	Atiya	5437
754	Gabbi	Matar	6543
754	Gabbi	Matar	564

אין אפשרות להציג את התמונה.

2NF

- Table must be in 1NF
- Non-prime attributes do not depend on a strict(proper) subset of a candidate key.

But StudentFirst, StudentLast and Address depend only on StudentId

What is the key?

StudentId+CourseId

StudentId	StudentFirst	StudentLast	Address	CourseId	Grade
542	Yossi	Agasi	Harambam 45, Ariel	4244	87
542	Yossi	Agasi	Harambam 45, Ariel	3423	65
956	Tamar	Atiya	Hadekel 12, Herzeliya	4244	86
542	Yossi	Agasi	Harambam 45, Ariel	6734	80

Fixing Table to Become 2NF

- In order to correct a relation that is not in 2NF, we split the information into 2 tables:

StudentId	StudentFirst	StudentLast	Address	CourseId	Grade
542	Yossi	Agasi	Harambam 45, Ariel	4244	87
542	Yossi	Agasi	Harambam 45, Ariel	3423	65
956	Tamar	Atiya	Hadekel 12, Herzeliya	4244	86
542	Yossi	Agasi	Harambam 45, Ariel	6734	80

StudentId	StudentFirst	StudentLast	Address
542	Yossi	Agasi	Harambam 45, Ariel
956	Tamar	Atiya	Hadekel 12, Herzeliya

StudentId	CourseId	Grade
542	4244	87
542	3423	65
956	4244	86
542	6734	80

Note that the new tables have 20 cells in total, while the original table had 24 cells. The new tables have 105 characters (combined) while the old table had 143.

2NF (cont.)

- Given: R(author, bookId, #pages)
- Each book can have one or more authors
- What is the candidate key?
  - {author, bookId}
- Is it in 2NF?
  - No:
    - bookId → #pages
    - {bookId} isn't a key
- How to fix?
  - Split to R1(author, bookId) and R2(bookId, #pages)

Authors Relation

Books Relation

3NF

except trivial

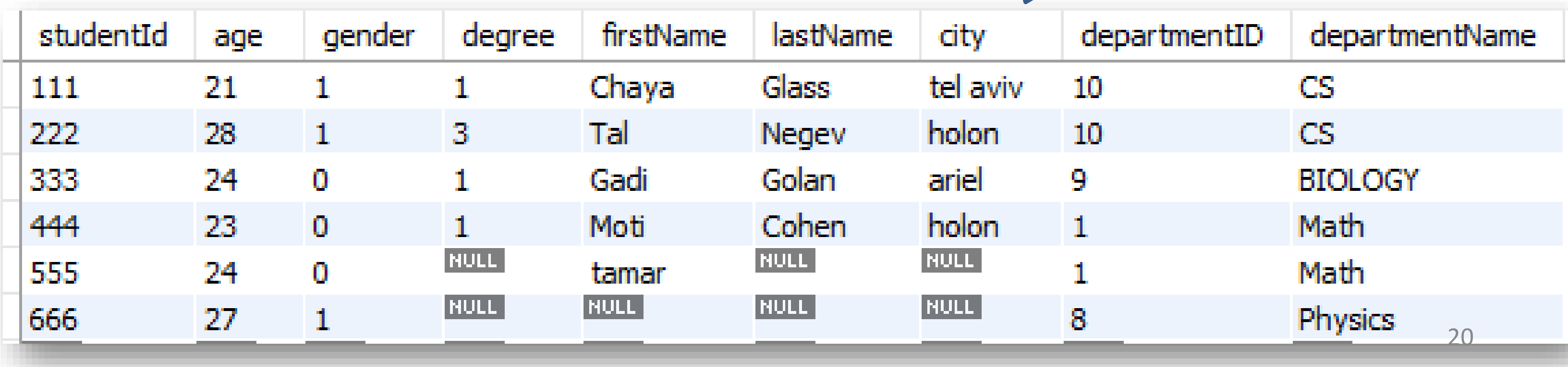
- Table must be in 2NF
- Non-prime attributes cannot depend on any set that isn't a super-key (transitive dependency).

But departmentName depends only on departmentId

What is the key?

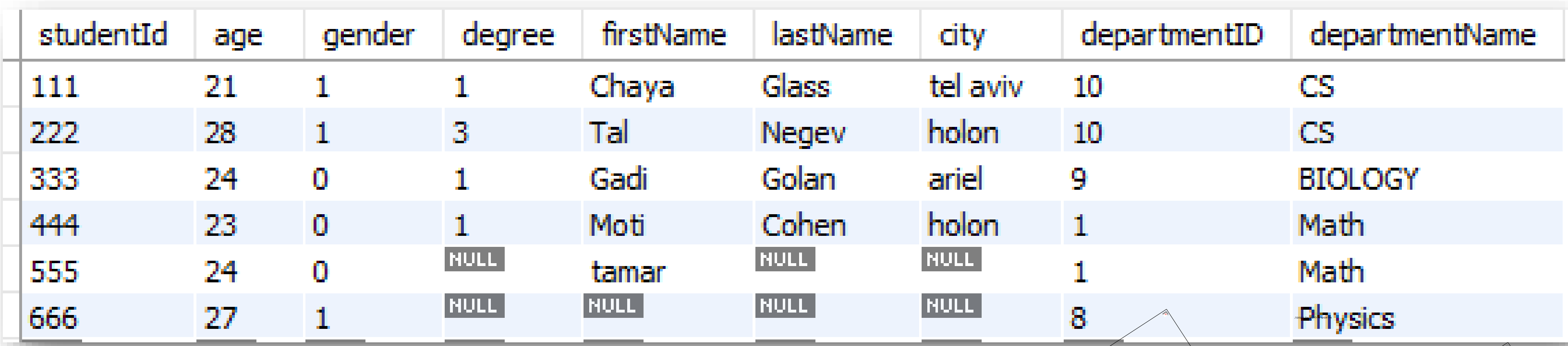
StudentId

Is it 2Nf?

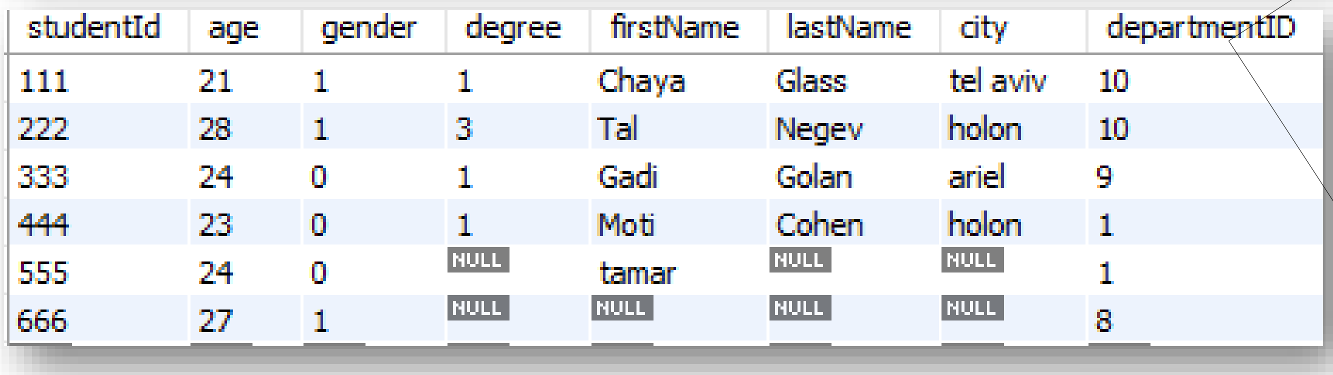


studentId	age	gender	degree	firstName	lastName	city	departmentID	departmentName
111	21	1	1	Chaya	Glass	tel aviv	10	CS
222	28	1	3	Tal	Negev	holon	10	CS
333	24	0	1	Gadi	Golan	ariel	9	BIOLOGY
444	23	0	1	Moti	Cohen	holon	1	Math
555	24	0	NULL	tamar	NULL	NULL	1	Math
666	27	1	NULL	NULL	NULL	NULL	8	Physics

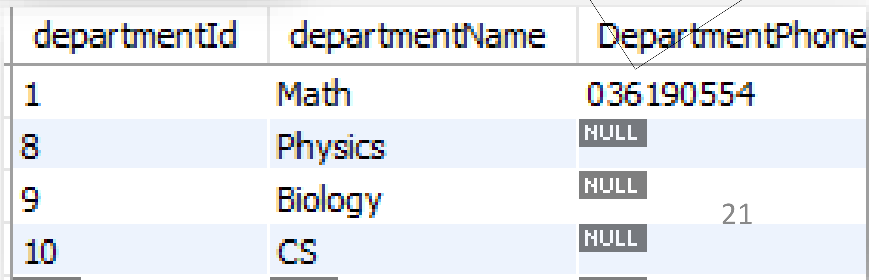
Fixing Table to Become 3NF



studentId	age	gender	degree	firstName	lastName	city	departmentID	departmentName
111	21	1	1	Chaya	Glass	tel aviv	10	CS
222	28	1	3	Tal	Negev	holon	10	CS
333	24	0	1	Gadi	Golan	ariel	9	BIOLOGY
444	23	0	1	Moti	Cohen	holon	1	Math
555	24	0	NULL	tamar	NULL	NULL	1	Math
666	27	1	NULL	NULL	NULL	NULL	8	Physics



studentId	age	gender	degree	firstName	lastName	city	departmentID
111	21	1	1	Chaya	Glass	tel aviv	10
222	28	1	3	Tal	Negev	holon	10
333	24	0	1	Gadi	Golan	ariel	9
444	23	0	1	Moti	Cohen	holon	1
555	24	0	NULL	tamar	NULL	NULL	1
666	27	1	NULL	NULL	NULL	NULL	8



departmentId	departmentName	DepartmentPhone
1	Math	036190554
8	Physics	NULL
9	Biology	NULL
10	CS	NULL

Boyce and Codd Normal Form (BCNF)

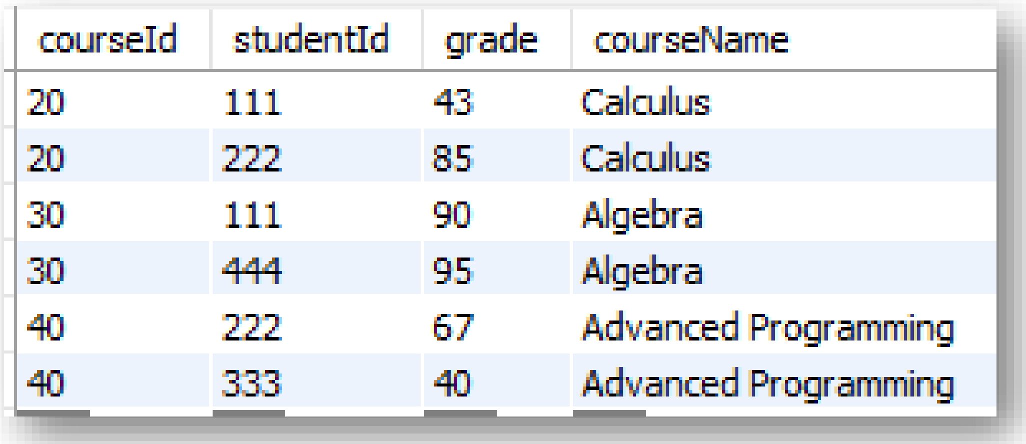
- BCNF is sometimes referred to as 3.5NF.
- Table must be in 3NF.
- For any two sets, X, Y, ($Y \nsubseteq X$) such that X→Y, X is a super-key.
- Note: If Y is prime, and X→Y, and X is not a super-key, while the table might be in 3NF, it is not in BCNF.

BCNF (3.5NF) example

CK ?
{courseId, studentId}
{courseName, studentId}

Dependencies ?
courseId -> courseName
courseName -> courseId
studentId -> studentName
courseId, studentId -> grade
courseName, studentId -> grade

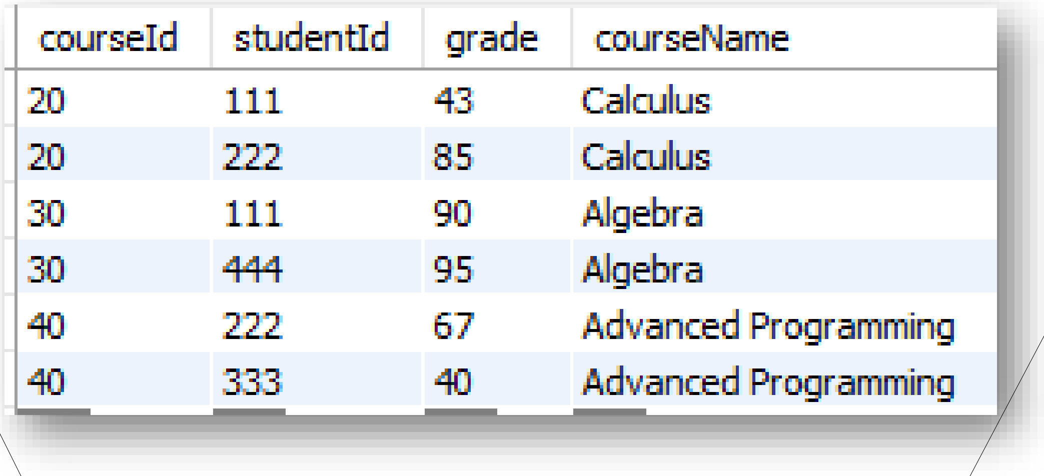
Grades



courseId	studentId	grade	courseName
20	111	43	Calculus
20	222	85	Calculus
30	111	90	Algebra
30	444	95	Algebra
40	222	67	Advanced Programming
40	333	40	Advanced Programming

2NF?
– yes, “grade” is the only non-prime attribute and it does not depend on a (strict/proper) subset of a candidate key
3NF?
– yes, “grade” is the only non-prime attribute and it depends only on one of the super key
BCNF?
– no ,courseName -> courseId , and courseId is not a super key

Fixing Table to Become 3.5NF



courseId	studentId	grade	courseName
20	111	43	Calculus
20	222	85	Calculus
30	111	90	Algebra
30	444	95	Algebra
40	222	67	Advanced Programming
40	333	40	Advanced Programming

courseId	studentId	grade

courseId	courseName

Boyce and Codd Normal Form (BCNF)

- True or false?
  - Any 3NF relation with a single candidate key is also in BCNF.

True:

Let a relation with two sets of attributes: X,Y so that X->Y.

If Y is non-prime then from 3NF we get that X is a super-key.
If Y is prime, assume by contradiction that X is not a super-key, if we replace Y with X in Y's candidate key (and minimize) we get a second candidate key (since $Y \nsubseteq X$)

1-3.5NF

- The data depends on the key (2NF), the whole key (3NF) and nothing but the key (3.5NF)

"No, I do not think 'The truth, the whole truth, and nothing but the truth' is overkill."

BCNF (cont.)

- Look at the following table used in a mobile company:
- R(mobilePhoneNum, simSerialNumber, callDateTime)
- Assumptions:
  - Once a phone number is burnt into a SIM card it can't be changed
  - mobilePhoneNum can be burnt on more than one sim card
- Dependencies:
  - simSerialNumber → mobilePhoneNum
  - {callDateTime, mobilePhoneNum} → simSerialNumber
- Candidate-keys:
  - {callDateTime, simSerialNumber }
  - {callDateTime, mobilePhoneNum}
- 2NF? 3NF?
- Is it BCNF?
- simSerialNumber → mobilePhoneNum, but {simSerialNumber} is not a super-key.

4NF

- Look at the following table:
  Each team player represents
  his department/s.

StudentId	Department	SportTeam
111	CS	Soccer
111	Biology	Soccer
222	Biology	Basketball
222	Biology	Soccer
333	CS	Basketball

StudentId	Department	SportTeam
111	CS	Soccer
111	Biology	Soccer
111	CS	Baseball
111	Biology	Baseball
222	Biology	Basketball
222	Biology	Soccer
333	CS	Basketball

- The key is:
  - {studentId, department, sportTeam}
- It doesn't violate NF 1-3.5
- But still it seems wrong:
  - What happens if 111 joins another sportTeam?
  - What happens if 222 joins another department?

4NF (cont.)

- 4NF requires BCNF + no multivalued dependencies.
- A multivalued dependency occurs when the presence of one or more rows in a table implies the presence of one or more other rows in that same table.
- That is, from observing some *rows,* one can deduce the presence of other rows.

StudentId	Department	SportTeam
111	CS	Soccer
111	Biology	Soccer
111	CS	Baseball

StudentId	Department	SportTeam
111	CS	Soccer
111	Biology	Soccer
111	CS	Baseball
111	Biology	Baseball

4NF (cont.)

- In our example, both the sportTeam and the department are independent of each-other, but both are multivalued dependent on studentId.
- We write this as:
  - studentId -->> department
  - studentId -->> sportTeam
- Every table should hold a single "idea" or "theme"!

Multivalued Dependency (Formal Definition)

- Multi dependency is a condition on the existence of rows (entries / tuples / entities) in the relation.
- Given two sets of attributes, A, and B, we say that A multidetermines B (A -->> B) if:
  - Let C = R \ (A U B) (that is, all the rest of the attributes)
  - Given rows x and y, such that:
    - x[A] = y[A] and
    - x[B] ≠ y[B] and
    - x[C] ≠ y[C]
  - Entails that, there exists a row z, such that:
    - z[A] = x[A] (= y[A]) and
    - z[B] = x[B] and
    - z[C] = y[C]

	A	B	C
x	a1	b1	c1
y	a1	b2	c2
z	a1	b1	c2
w			

5NF

- 5NF is related to situations in which some rules are applied on the rows of the table.
- In such situations, if the table can be decomposed into smaller tables by removing redundant data, the table is not in 5NF.
- "Only in rare situations does a 4NF table not conform to the higher normal form 5NF. These are situations in which a complex real-world constraint governing the valid combinations of attribute values in the 4NF table is not implicit in the structure of that table." (Wikipedia)
- Therefore, we won't be dealing with 5NF.

Question

- Let us look on the following relation:
  - R(A,B,C,D)
  - {A,B}→D
  - {A,D}→C

If we have an attribute that appears only on the right of the dependency list, what may we conclude?

- What are the candidate key(s)?
  - C.k: {A,B}

If we have an attribute that does not appear on the right of the dependency list, what may we conclude?

- Is it in 2NF?
  - yes. Non prime attributes don't depend on subset of the candidate key.
- Is it in 3NF?
  - No. C is non prime and depents on {A,D} which is not super key.

THANK YOU FOR LISTENING TO MY PRESENTATION!!!
ANY QUESTIONS?