Boyce-Codd Normal Form (BCNF)

- When a relation has more than one candidate key, anomalies may result even though the relation is in 3NF.
- 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keys
- i.e. composite candidate keys with at least one attribute in common.
- BCNF is based on the concept of a *determinant*.
- A determinant is any attribute (simple or composite) on which some other attribute is fully functionally dependent.
- A relation is in BCNF is, and only if, every determinant is a candidate key.

Consider the following relation and determinants.

R($\underline{a,b}$,c,d)
a,c -> b,d
a,d -> b

Here, the first determinant suggests that the primary key of R could be changed from a,b to a,c. If this change was done all of the non-key attributes present in R could still be determined, and therefore this change is legal. However, the second determinant indicates that a,d determines b, but a,d could not be the key of R as a,d does not determine all of the non key attributes of R (it does not determine c). We would say that the first determinate is a candidate key, but the second determinant is not a candidate key, and thus this relation is not in BCNF (but is in 3rd normal form).

Normalisation to BCNF - Example 1

Patient No	Patient Name	Appointment Id	Time	Doctor
1	John	0	09:00	Zorro
2	Kerr	0	09:00	Killer
3	Adam	1	10:00	Zorro
4	Robert	0	13:00	Killer
5	Zane	1	14:00	Zorro

Lets consider the database extract shown above. This depicts a special dieting clinic where the each patient has 4 appointments. On the first they are weighed, the second they are exercised, the third their fat is removed by surgery, and on the fourth their mouth is stitched closed… Not all patients need all four appointments! If the Patient Name begins with a letter before "P" they get a morning appointment, otherwise they get an afternoon appointment. Appointment 1 is either 09:00 or 13:00, appointment 2 10:00 or 14:00, and so on. From this (hopefully) make-believe scenario we can extract the following determinants:

DB(Patno,PatName,appNo,time,doctor)

Patno -> PatName
Patno,appNo -> Time,doctor
Time -> appNo

Now we have to decide what the primary key of DB is going to be. From the information we have, we could chose:

DB(Patno,PatName,appNo,time,doctor) (example 1a)

or

DB(Patno,PatName,appNo,time,doctor) (example 1b)

Example 1a - DB(Patno,PatName,appNo,time,doctor)

- 1NF Eliminate repeating groups.

None:

DB(Patno,PatName,appNo,time,doctor)

- 2NF Eliminate partial key dependencies

```
DB(Patno,appNo,time,doctor)
R1(Patno,PatName)
```

- 3NF Eliminate transitive dependencies

```
None: so just as 2NF
```

- BCNF Every determinant is a candidate key
  DB(Patno,appNo,time,doctor)
  R1(Patno,PatName)
- Go through all determinates where ALL of the left hand attributes are present in a relation and at least ONE of the right hand attributes are also present in the relation.
- Patno -> PatName
  Patno is present in DB, but not PatName, so not relevant.
- Patno,appNo -> Time,doctor
  All LHS present, and time and doctor also present, so relevant. Is this a candidate key? Patno,appNo IS the key, so this is a candidate key. Thus this is OK for BCNF compliance.
- Time -> appNo
  Time is present, and so is appNo, so relevant. Is this a candidate key. If it was then we could rewrite DB as:
  DB(Patno,appNo,time,doctor)
  This will not work, as you need both time and Patno together to form a unique key. Thus this determinate is not a candidate key, and therefore DB is not in BCNF. We need to fix this.
- BCNF: rewrite to
  DB(Patno,time,doctor)

R1(<u>Patno</u>,PatName)
R2(<u>time</u>,appNo)

time is enough to work out the appointment number of a patient. Now BCNF is satisfied, and the final relations shown are in BCNF.

Example 1b - DB(<u>Patno</u>,PatName,appNo,<u>time</u>,doctor)

- 1NF Eliminate repeating groups.

None:

DB(<u>Patno</u>,PatName,appNo,<u>time</u>,doctor)

- 2NF Eliminate partial key dependencies

```
DB(Patno,time,doctor)
R1(Patno,PatName)
R2(time,appNo)
```

- 3NF Eliminate transitive dependencies

```
None: so just as 2NF
```

- BCNF Every determinant is a candidate key
- `DB(Patno,time,doctor)`
- `R1(Patno,PatName)`
- `R2(time,appNo)`
- Go through all determinates where ALL of the left hand attributes are present in a relation and at least ONE of the right hand attributes are also present in the relation.
- Patno -> PatName
  Patno is present in DB, but not PatName, so not relevant.
- Patno,appNo -> Time,doctor
  Not all LHS present, so not relevant.
- Time -> appNo
  Time is present, and so is appNo, so relevant. This is a candidate key. However, Time is currently the key for R2, so satisfies the rules for BCNF.
- BCNF: as 3NF
  DB(<u>Patno,time</u>,doctor)
  R1(<u>Patno</u>,PatName)
  R2(<u>time</u>,appNo)

Summary - Example 1

This example has demonstrated three things:

- BCNF is stronger than 3NF, relations that are in 3NF are not necessarily in BCNF
- BCNF is needed in certain situations to obtain full understanding of the data model
- there are several routes to take to arrive at the same set of relations in BCNF.

- Unfortunately there are no rules as to which route will be the easiest one to take.

Example 2

```
Grade_report(StudNo,StudName,(Major,Adviser,
 (CourseNo,Ctitle,InstrucName,InstructLocn,Grade)))
```

- Functional dependencies

```
 StudNo -> StudName
 CourseNo -> Ctitle,InstrucName
 InstrucName -> InstrucLocn
 StudNo,CourseNo,Major -> Grade
 StudNo,Major -> Advisor
 Advisor -> Major
```

- Unnormalised

```
Grade_report(StudNo,StudName,(Major,Advisor,
 (CourseNo,Ctitle,InstrucName,InstructLocn,Grade)))
```

- 1NF Remove repeating groups

```
Student(StudNo,StudName)
StudMajor(StudNo,Major,Advisor)
StudCourse(StudNo,Major,CourseNo,
  Ctitle,InstrucName,InstructLocn,Grade)
```

- 2NF Remove partial key dependencies

```
Student(StudNo,StudName)
StudMajor(StudNo,Major,Advisor)
StudCourse(StudNo,Major,CourseNo,Grade)
Course(CourseNo,Ctitle,InstrucName,InstructLocn)
```

- 3NF Remove transitive dependencies

```
Student(StudNo,StudName)
StudMajor(StudNo,Major,Advisor)
StudCourse(StudNo,Major,CourseNo,Grade)
Course(CourseNo,Ctitle,InstrucName)
Instructor(InstructName,InstructLocn)
```

- BCNF Every determinant is a candidate key
- Student : only determinant is StudNo
- StudCourse: only determinant is StudNo,Major
- Course: only determinant is CourseNo
- Instructor: only determinant is InstrucName
- StudMajor: the determinants are
- StudNo,Major, or
- Adviser

Only StudNo,Major is a candidate key.

- BCNF

```
Student(StudNo,StudName)
StudCourse(StudNo,Major,CourseNo,Grade)
Course(CourseNo,Ctitle,InstrucName)
Instructor(InstructName,InstructLocn)
StudMajor(StudNo,Advisor)
Adviser(Adviser,Major)
```

Problems BCNF overcomes

STUDENT	**MAJOR**	**ADVISOR**
123	PHYSICS	EINSTEIN
123	MUSIC	MOZART
456	BIOLOGY	DARWIN
789	PHYSICS	BOHR
999	PHYSICS	EINSTEIN

- If the record for student 456 is deleted we lose not only information on student 456 but also the fact that DARWIN advises in BIOLOGY
- we cannot record the fact that WATSON can advise on COMPUTING until we have a student majoring in COMPUTING to whom we can assign WATSON as an advisor.

In BCNF we have two tables:

STUDENT	**ADVISOR**
123	EINSTEIN
123	MOZART
456	DARWIN
789	BOHR
999	EINSTEIN

ADVISOR	**MAJOR**
EINSTEIN	PHYSICS
MOZART	MUSIC
DARWIN	BIOLOGY
BOHR	PHYSICS

Returning to the ER Model

- Now that we have reached the end of the normalisation process, you must go back and compare the resulting relations with the original ER model
- You may need to alter it to take account of the changes that have occurred during the normalisation process Your ER diagram should always be a prefect reflection of the model you are going to implement in the database, so keep it up to date!
- The changes required depends on how good the ER model was at first!

Normalisation Example

Library

Consider the case of a simple video library. Each video has a title, director, and serial number. Customers have a name, address, and membership number. Assume only one copy of each video exists in the library. We are given:

```
video(title,director,serial)
customer(name,addr,memberno)
hire(memberno,serial,date)

  title->director,serial
  serial->title
  serial->director
  name,addr -> memberno
  memberno -> name,addr
  serial,date -> memberno
```

What normal form is this?

- No repeating groups, so at least 1NF
- 2NF? There is a composite key in hire. Investigate further... Can memberno in hire be found with just serial or just date. NO. Therefore relation is in at least 2NF.
- 3NF? serial->director is a non-key dependency. Therefore the relations are currently in 2NF.

Convert from 2NF to 3NF.

```
Rewrite
   video(title,director,serial)
To
   video(title,serial)
   serial(serial,director)

Therefore the new relations become:
   video(title,serial)
   serial(serial,director)
```

```
customer(name,addr,memberno)
hire(memberno,serial,date)
```

In BCNF? Check if every determinant is a candidate key.

```
video(title,serial)
    Determinants are:
        title->director,serial  Candidate key
        serial->title           Candidate key
    video in BCNF

serial(serial,director)
    Determinants are:
        serial->director        Candidate key
    serial in BCNF

customer(name,addr,memberno)
    Determinants are:
       name,addr -> memberno    Candidate key
       memberno -> name,addr    Candidate key
    customer in BCNF

hire(memberno,serial,date)
    Determinants are:
       serial,date -> memberno   Candidate key
    hire in BCNF
```

Therefore the relations are also now in BCNF.

Reference: http://db.grussell.org/section009.html