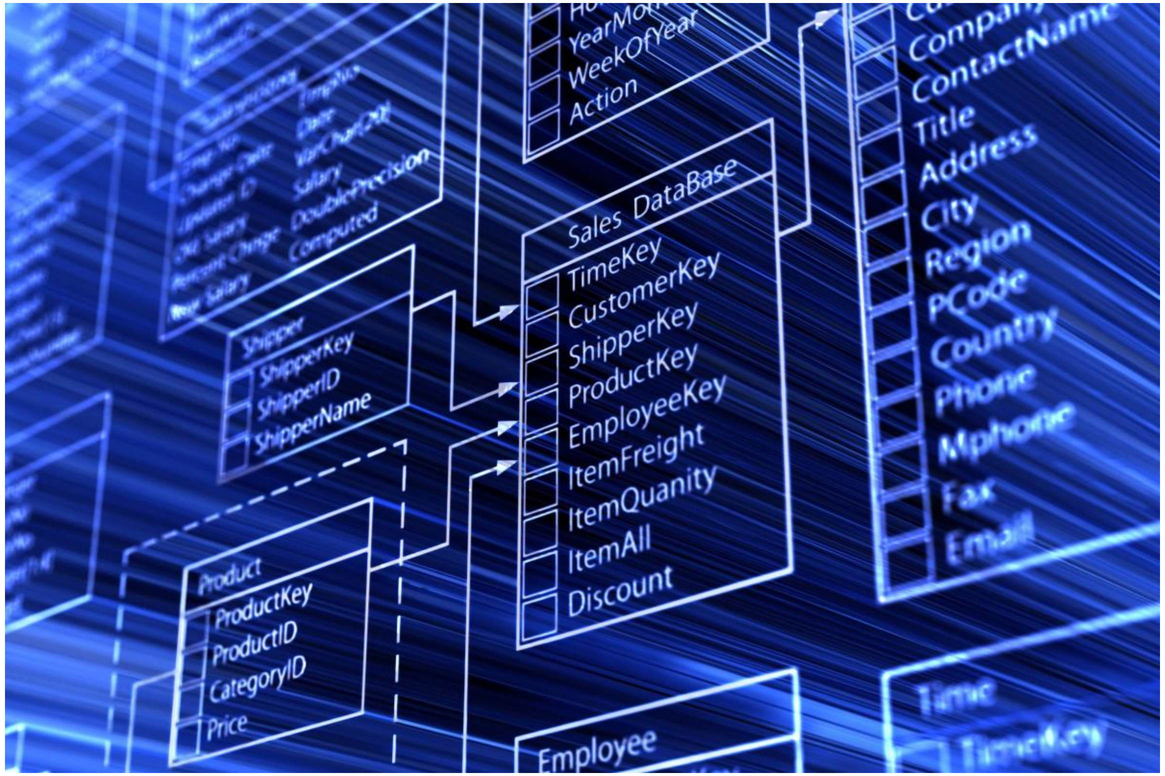


INTRODUCTION TO DATABASES USING ORACLE

420-983-VA

NORMALIZATION

NORMAL FORMS

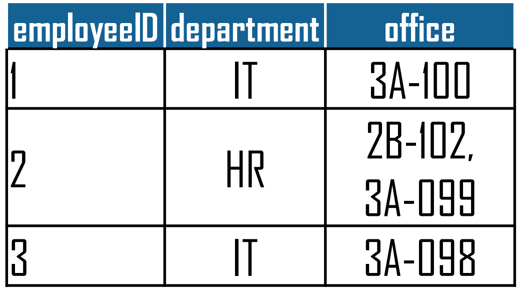
Given a relation schema, we need to decide whether it is a good design or we need to decompose it into smaller relations. Such a decision must be guided by an understanding of what problems, if any, arise from the current schema. To provide such guidance, several normal forms have been proposed. If a relation schema is in one of these normal forms, we know that certain kinds of problems cannot arise.

NORMAL FORMS

First Normal Form (1NF)

A relation is in first normal form if every field contains only atomic values, that is, no lists or sets or other composite types. This requirement is implicit in the definition of the relational model. Some of the newer database systems are relaxing this requirement.

Example of relation not in 1NF:



employeeID	department	office
1	IT	3A-100
2	HR	2B-102, 3A-099
3	IT	3A-098

Office columns contains a set of atomic values for employee with ID 2.

NORMAL FORMS

Decomposing First Normal Form (1NF)

Each element of the list is considered for each row where it occurred. Relation will not allow lists or sets for any of its attributes.

Example

employeeID	department	office
1	IT	3A-100
2	HR	2B-102, 3A-099
3	IT	3A-098

employeeID	department	office
1	IT	3A-100
2	HR	2B-102
2	HR	3A-099
3	IT	3A-098

NORMAL FORMS

Second Normal Form (2NF)

A database is in second normal form if for every FD $X \rightarrow A$ one of the following is true:

a) $X \rightarrow A$ is a trivial FD (i.e. $A \in X$), or.
b) X is a superkey, or
c) A is a subkey, or
d) X is not a subkey.

Example of relation that is not in 2NF:

Orders(productId, clientId, date, quantity, productName)

prodID	clientId	date	qty	prodName
4033	1	3 Jan 20	1	Lemon
4011	2	4 Mar 20	2	Banana
4051	1	9 Jun 20	1	Mango
4051	2	12 Jun 20	1	Mango

Primary key = {productId, clientId, date} and {productid} $\longrightarrow$ productName

NORMAL FORMS

Decomposing into Second Normal Form (2NF)

Decompose into 1NF, then each attribute that is functional dependent by subset of a key is removed from the relation and a new relation is created with the subset of the key and the removed attribute.

Example:

Orders(productId, clientId, date, quantity, productName)

prodID	clientId	date	qty	prodName
4033	1	3 Jan 20	1	Lemon
4011	2	4 Mar 20	2	Banana
4051	1	9 Jun 20	1	Mango
4051	2	12 Jun 20	1	Mango

2NF

Orders(productId, clientId, date, quantity)

Products(productId, productName)

prodID	clientId	date	qty
4033	1	3 Jan 20	1
4011	2	4 Mar 20	2
4051	1	9 Jun 20	1
4051	2	12 Jun 20	1

prodID	prodName
4033	Lemon
4011	Banana
4051	Mango

NORMAL FORMS

Boyce-Codd Normal Form (BCNF)

A database is in the Boyce-Codd normal form if for any FD X→A (X set of attributes, A one attribute – recall FD splitting) we have one of the following:

a) X→A is a trivial FD (i.e. $A \in X$), or.

b) X is a superkey.

NORMAL FORMS

Boyce-Codd Normal Form (BCNF)

Example:

Consider following relation:

Grading(studentId, subjectCode, subjectName, exam#, score, grade)

Which are the keys?

NORMAL FORMS

Boyce-Codd Normal Form (BCNF)

Example:

Consider following relation:

Grading(studentId, subjectCode, subjectName, exam#, score, grade)

KEYS

{studentId, subjectCode, exam#}

{studentId, subjectName, exam#}

NORMAL FORMS

Boyce-Codd Normal Form (BCNF)

Example:

Consider following relation:

Grading(studentId, subjectCode, subjectName, exam#, score, grade)

KEYS

How about FDs?

{studentId, subjectCode, exam#}

{studentId, subjectName, exam#}

NORMAL FORMS

Boyce-Codd Normal Form (BCNF)

Example:

Consider following relation:

Grading(studentId, subjectCode, subjectName, exam#, score, grade)

KEYS

{studentId, subjectCode, exam#}

{studentId, subjectName, exam#}

Functional Dependencies (FDs)

subjectCode ⟶ subjectName

subjectName ⟶ subjectCode

studentId, subjectCode, exam# ⟶ score

studentId, subjectName, exam# ⟶ grade

score ⟶ grade

… more other trivial and key based FDs

NORMAL FORMS

Boyce-Codd Normal Form (BCNF)

Example:

Consider following relation:

Grading(studentId, subjectCode, subjectName, exam#, score, grade)

KEYS

{studentId, subjectCode, exam#}

{studentId, subjectName, exam#}

Functional Dependencies (FDs)

subjectCode → subjectName

subjectName → subjectCode

~~studentId, subjectCode, exam# → score~~

~~studentId, subjectName, exam# → grade~~

Key FDs

score → grade

… more other trivial and key based FDs

NORMAL FORMS

Boyce-Codd Normal Form (BCNF)

Example:

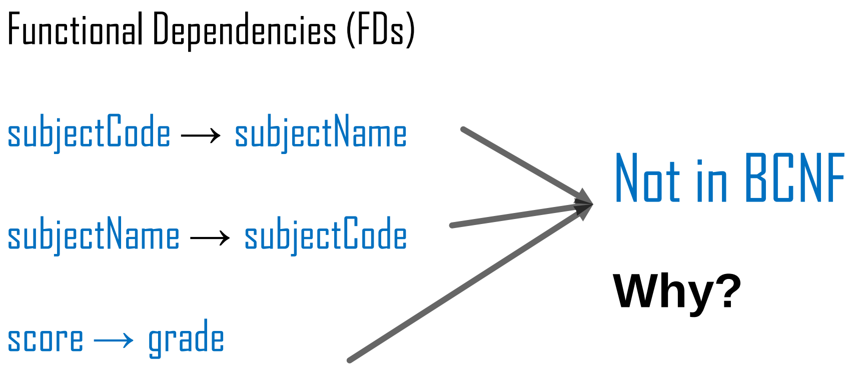
Consider following relation:

Grading(studentId, subjectCode, subjectName, exam#, score, grade)

KEYS

{studentId, subjectCode, exam#}

{studentId, subjectName, exam#}



Functional Dependencies (FDs)

subjectCode ⟶ subjectName

subjectName ⟶ subjectCode

score ⟶ grade

Not in BCNF

Why?

NORMAL FORMS

Boyce-Codd Normal Form (BCNF)

Example:

Consider following relation:

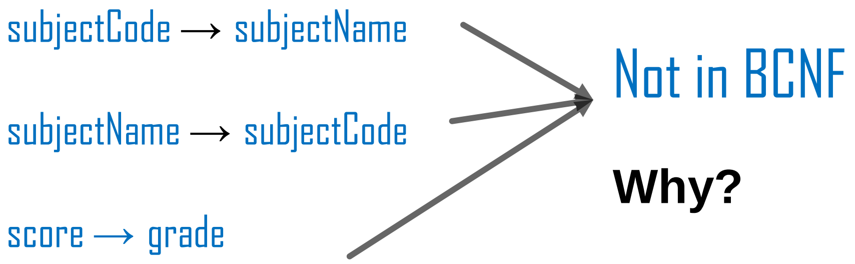
Grading(studentId, subjectCode, subjectName, exam#, score, grade)

KEYS

{studentId, subjectCode, exam#}

{studentId, subjectName, exam#}

Functional Dependencies (FDs)



subjectCode → subjectName

subjectName → subjectCode

score → grade

Not in BCNF

Why?

Remember (in BCNF):

∀FD X→A we have:

a) X→A is a trivial FD (i.e. A∈X), or.

b) X is a superkey.

NORMAL FORMS

Decomposing into Boyce-Codd Normal Form (BCNF)

Repeatedly apply decomposition until all resulted relations are in BCNF. That is for each non-trivial FD $X \rightarrow A$ that violates BCNF for relation R we decompose R in two relations:

1. R_0 containing attributes in X and all attributes from R that are not in X^+.
2. R_1 containing all attributes from X^+.

NORMAL FORMS

Dependency Preservation in Boyce-Codd Normal Form (BCNF)

Consider relation: Bookings(movieTitle, theater, city)
We assume that each movie is playing in a single theater in a given city. Each theater may show multiple movies (multiplex). Theater names are unique.

For this relation we have following FD's:

theater ⟶ city

movieTitle, city ⟶ theater

Can you find the keys?

NORMAL FORMS

Dependency Preservation in Boyce-Codd Normal Form (BCNF)

Consider relation: Bookings(movieTitle, theater, city)
We assume that each movie is playing in a single theater in a given city. Each theater may show multiple movies (multiplex). Theater names are unique.

For this relation we have following FD's:

theater ⟶ city

movieTitle, city ⟶ theater

Keys:

{movieTitle, city}
{movieTitle, theater}

NORMAL FORMS

Dependency Preservation in Boyce-Codd Normal Form (BCNF)

Consider relation: Bookings(movieTitle, theater, city)
We assume that each movie is playing in a single theater in a given city. Each theater may show multiple movies (multiplex). Theater names are unique.

For this relation we have following FD’s:

theater ⟶ city ⟶ Violates BCNF

movieTitle, city ⟶ theater

Keys:

{movieTitle, city}
{movieTitle, theater}

NORMAL FORMS

Dependency Preservation in Boyce-Codd Normal Form (BCNF)

Consider relation: Bookings(movieTitle, theater, city)
We assume that each movie is playing in a single theater in a given city. Each theater may show multiple movies (multiplex). Theater names are unique.

For this relation we have following FD's:

theater ⟶ city ⟶ Violates BCNF

movieTitle, city ⟶ theater

Decompose Bookings into:

Theaters(theater, city)
Bookings(movieTitle, theater)

NORMAL FORMS

Dependency Preservation in Boyce-Codd Normal Form (BCNF)

Consider relation: Bookings(movieTitle, theater, city)
We assume that each movie is playing in a single theater in a given city. Each theater may show multiple movies (multiplex). Theater names are unique.

For this relation we have following FD’s:

theater → city

movieTitle, city → theater ⟶ Is this FD satisfied?

Decompose Bookings into:

Theaters(theater, city)
Bookings(movieTitle, theater)

NORMAL FORMS

Dependency Preservation in Boyce-Codd Normal Form (BCNF)

Consider relation: Bookings(movieTitle, theater, city)
We assume that each movie is playing in a single theater in a given city. Each theater may show multiple movies (multiplex). Theater names are unique.

For this relation we have following FD's:

theater ⟶ city

movieTitle, city ⟶ theater ⟶ Is this FD satisfied? ✗

Decompose Bookings into:

Theaters(theater, city)
Bookings(movieTitle, theater)

NORMAL FORMS

Dependency Preservation in Boyce-Codd Normal Form (BCNF)

Consider relation: Bookings(movieTitle, theater, city)
We assume that each movie is playing in a single theater in a given city. Each theater may show multiple movies (multiplex). Theater names are unique.

For this relation we have following FD's:

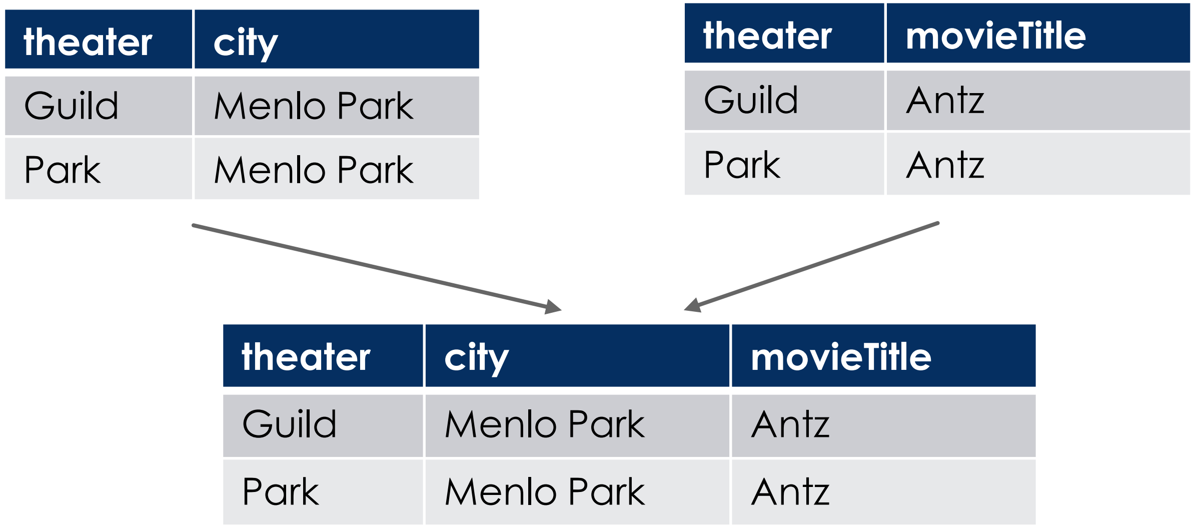
theater → city

movieTitle, city → theater

Decompose Bookings into:

Theaters(theater, city)
Bookings(movieTitle, theater)

Based on the decomposition the following two relations are allowed.



theater	city
Guild	Menlo Park
Park	Menlo Park

theater	movieTitle
Guild	Antz
Park	Antz

theater	city	movieTitle
Guild	Menlo Park	Antz
Park	Menlo Park	Antz

But this violates our initial FD assumption movieTitle, city → theater

NORMAL FORMS

Third Normal Form (3NF)

The solution for the dependency preservation problem observed before is to relax BCNF. A relation is in third normal form (3NF) if for any FD X→A we have one of the following:

a) X→A is a trivial FD (i.e. $A \in X$), or.
b) X is a superkey, or
c) A is a member of some key.

NORMAL FORMS

Third Normal Form (3NF)

The solution for the dependency preservation problem observed before is to relax BCNF. A relation is in third normal form (3NF) if for any FD X→A we have one of the following:

a) X→A is a trivial FD (i.e. A∈X), or.
b) X is a superkey, or
c) A is a member of some key.

city is part of a key, thus does not violate 3NF

In this case Bookings(movieTitle, theater, city) with

theater → city

movieTitle, city → theater

is in 3NF.

NORMAL FORMS

Third Normal Form (3NF)

Example:

Consider relation for regional list of telephone numbers: TEL(place, area, number where place represents the location (e.g. town) for this phone. Area represents the area code and number the phone number.

For this we have the following FD's:

area, number ⟶ place

place ⟶ area

In this case TEL(place, area, number) is in 3NF

NORMAL FORMS

Third Normal Form (3NF)

Example:

Consider relation for regional list of telephone numbers: TEL(place, area, number where place represents the location (e.g. town) for this phone. Area represents the area code and number the phone number.

For this we have the following FD's:

area, number ⟶ place

place ⟶ area

In this case TEL(place, area, number) is in 3NF

But does not preserve place ⟶ area , i.e. we may add records with the same place but different area codes.

NORMAL FORMS

Third Normal Form (3NF)

Example:

Consider relation for regional list of telephone numbers: TEL(place, area, number where place represents the location (e.g. town) for this phone. Area represents the area code and number the phone number.

For this we have the following FD's:

area, number → place

place → area

If we decompose it (according to BCNF) in TEL1(place, area) and TEL2(place, number)

Does not preserve FD area, number → place

How should we decompose this relation?

NORMAL FORMS

Third Normal Form (3NF)

Example:

Consider relation for regional list of telephone numbers: TEL(place, area, number where place represents the location (e.g. town) for this phone. Area represents the area code and number the phone number.

For this we have the following FD's:

area, number → place

place → area

TEL1(place, area, number) TEL2(place, area)