Relational Model

Complete Summary

Instructor:

Deepak Poonia

MTech, IISc Bangalore

GATE CSE AIR 53; AIR 67;
AIR 107; AIR 206; AIR 256

DBMS Complete Course:
https://www.goclasses.in/courses/Database-Management-Systems

NOTE :

Complete Discrete Mathematics & Complete Engineering Mathematics Courses, by GO Classes, are **FREE** for ALL learners.

Visit here to watch : https://www.goclasses.in/s/store/

SignUp/Login on Goclasses website for free and start learning.

GO CLASSES

GO Classes

GO CLASSES

FREE

COMPLETE COURSE

Engineering Mathematics

GATE CS IT

Freely Available Features

FREE

- All Video Lectures
- Quizzes
- Homeworks, Practice Sets
- Annotated Notes
- Toppers' Hand Written Notes
- GATE PYQs Video Solutions

SACHIN MITTAL SIR

www.goclasses.in

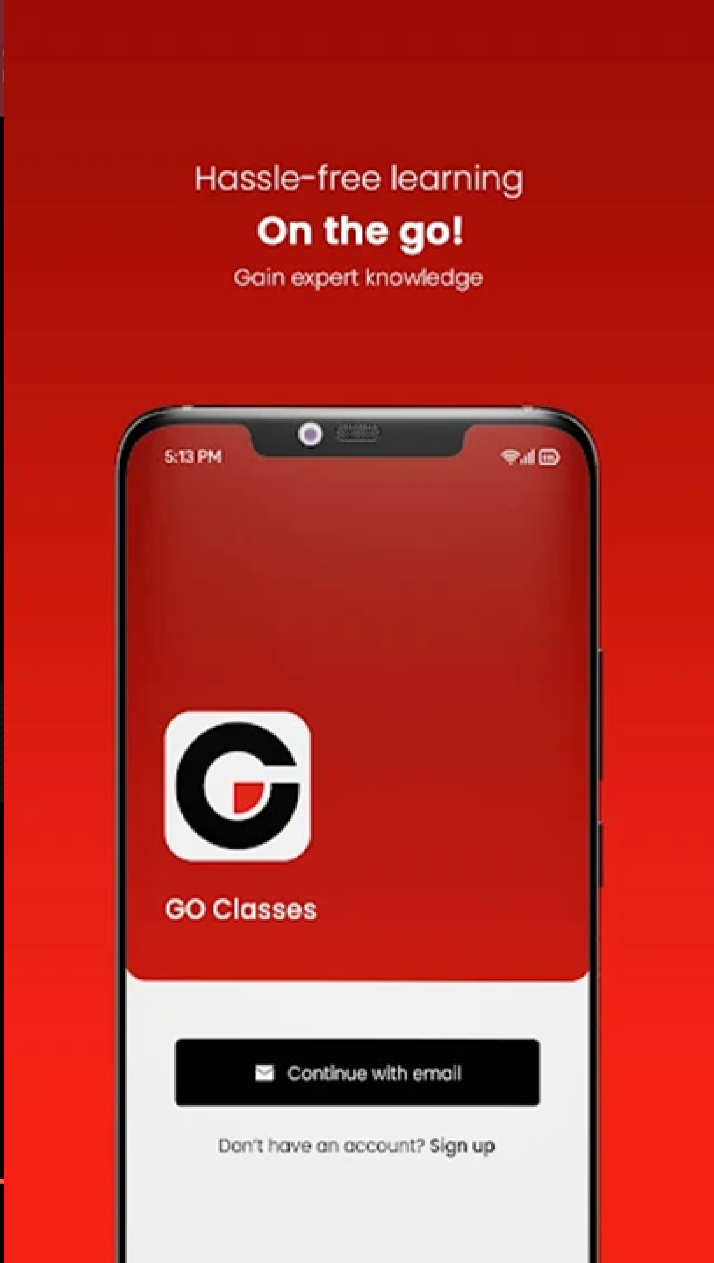
/ GoClasses

GO CLASSES
GO Classes
GO CLASSES
FREE
COMPLETE COURSE
Discrete Mathematics
GATE CS IT
Freely Available Features
FREE
All Video Lectures
Quizzes, Homeworks
Annotated Notes
Toppers' Hand Written Notes
Summary Lectures
GATE PYQs Video Solutions
DEEPAK POONIA SIR
www.goclasses.in
/ GoClasses

Download the GO Classes Android/ iOS App:

https://play.google.com/store/apps/details?id=com.goclasses.courses

Search “GO Classes” on Play Store / App Store.



We are on Telegram. **Contact us** for any help.

Link in the Description!!

Join GO Classes **Doubt Discussion** Telegram Group :

Username:

@GATECSE_GOCLASSES

We are on Telegram. **Contact us** for any help.

Join GO Classes Telegram Channel, Username: **@GOCLASSES_CSE**

Join GO Classes **Doubt Discussion** Telegram Group :

Username: **@GATECSE_Goclasses**

(Any doubt related to Goclasses Courses can also be asked here.)

Join GATEOverflow Doubt Discussion Telegram Group :

Username: **@GateOverflow_CSE**


CLASSES

Relational Model

Complete Summary

DBMS Summary:

1. Introduction to DBMS
2. Introduction to Relational Data Model

Database & DBMS

Database:

It is a large collection of interrelated data.

Example:

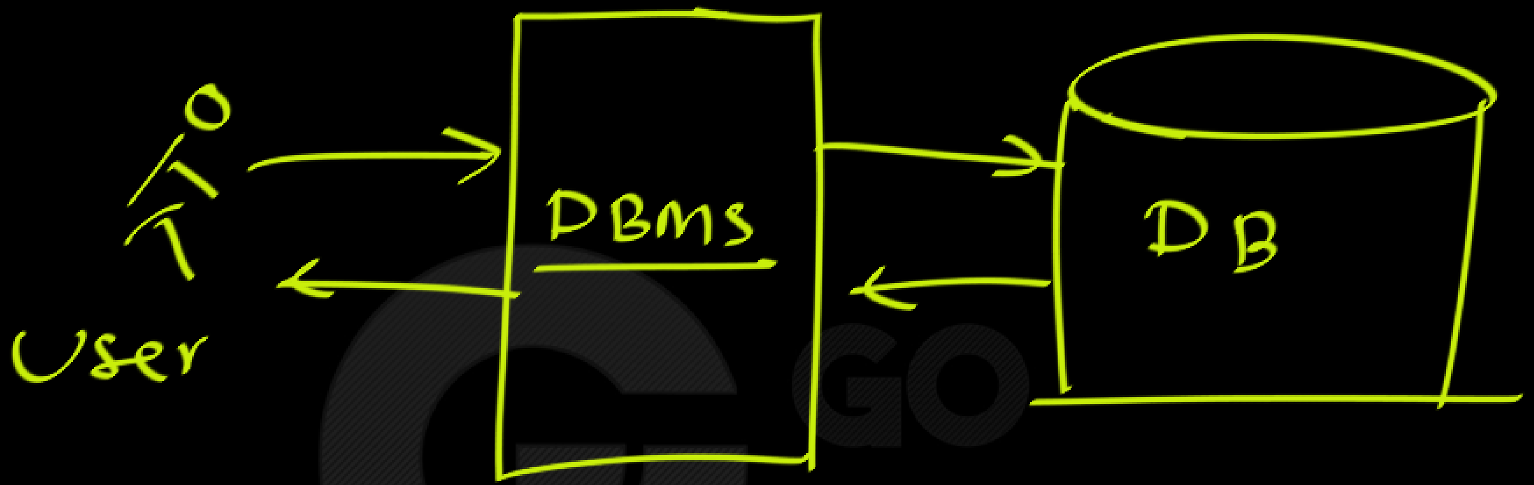
Customer database consists the fields as cname, cno, and ccity

Cname	Cno	Ccity

Database Management System (DBMS) :

- A DBMS is a software package designed to store and manage databases.

It is a collection of programs that enables user to create and maintain a database. A DBMS serves as an interface between the database and its end users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized.


User
DBMS
DB

What is a database management system?

a) A collection of data stored on a computer

b) A collection of related data representing facts that have meaning in the real world

c) A software system allowing users to create and maintain a database

d) A software system allowing users to define web-portal interfaces that interact with data files

What is a database management system?

Database

a) A collection of data stored on a computer

b) A collection of related data representing facts that have meaning in the real world

c) A software system allowing users to create and maintain a database

d) A software system allowing users to define web-portal interfaces that interact with data files

The Relational Model

for Databases

A Table

Table name

Product:

Column names

Name	Price	Category	Manufacturer
gizmo	$19.99	gadgets	GizmoWorks
Power gizmo	$29.99	gadgets	GizmoWorks
SingleTouch	$149.99	photography	Canon
MultiTouch	$203.99	household	Hitachi

Rows

The relational model is very simple and elegant; a database is a collection of one or more *relations*, where each relation is a table with rows and columns.

Table

Example of a Database

Student

StudName	StudNo	Class	Dept
Smith	17	1	CS
Brown	8	2	CS

Course

CourseName	CourseNo	Credits	Dept
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MA2410	3	MA
Database Management	CS3380	3	CS

Prerequisite

CourseNo	PrereqNo
CS3380	CS3320
CS3380	MA2410
CS3320	CS1310

Session

SessIdent	CourseNo	Semester	Year	Professor
85	MA2410	Fall	96	King
92	CS1310	Fall	96	Anderson
102	CS3320	Spring	97	Knuth
112	MA2410	Fall	97	Chang
119	CS1310	Fall	97	Anderson

GradeReport

StudNo	SessIdent	Grade
17	112	14
17	119	12
8	85	16
8	92	16
8	102	14

Basics of Relational Model

Relational Data Model:

The relational model gives us a single way to represent data: as a two-dimensional table called a relation.

Eg: The Movies Table

The rows each represent a movie, and the columns each represent a property of movies.

Column ≡ attribute ≡ field

title	*year*	*length*	*genre*
Gone With the Wind	1939	231	drama
Star Wars	1977	124	sciFi
Wayne's World	1992	95	comedy

a Tuple / Row / Record

an Entity

Figure 2.3: The relation Movies : Table Name

In the Relational Model Terminology:

A row is called a tuple.

Row = Record = Tuple = Entity

A column header is called an attribute.

Column Name = Attribute = Field

The table is called a relation.

Table = Relation = Entity set

Basics of Relational Model

Domain of an Attribute

```
Students(
   roll:  string,
   name:  string,
   age:  integer,
   gpa:  real
)
```

roll	name	age	gpa
0001	John	23	9.1
0002	Amit	24	8.7
0003	Vivek	25	9.2
0004	Sandhya	23	9.0
….	….	….	….

Relations and Attributes

- Each relation has some number of attributes
  - Sometimes called “columns”
- Each attribute has a domain
  - Specifies the set of valid values for the attribute
- The *account* relation:
  - 3 attributes
  - Domain of *balance* is the set of nonnegative integers
  - Domain of *branch_name* is the set of branch names in the bank

acct_id	branch_name	balance
A-301	New York	350
A-307	Seattle	275
A-318	Los Angeles	550
…	…	…

account

Domain of an attribute in Relational Model :

A domain D is a set of Atomic values.

By atomic we mean that each value in the domain is indivisible.

2.2.4 Domains

The relational model requires that each component of each tuple be atomic; that is, it must be of some elementary type such as integer or string. It is not permitted for a value to be a record structure, set, list, array, or any other type that reasonably can have its values broken into smaller components.

Basics of Relational Model

Schema Vs Instance

Intention

Extension

A Table / Relation

Table name

Product:

Column names

Schema

Name	Price	Category	Manufacturer
gizmo	$19.99	gadgets	GizmoWorks
Power gizmo	$29.99	gadgets	GizmoWorks
SingleTouch	$149.99	photography	Canon
MultiTouch	$203.99	household	Hitachi

Instance

Rows

Relational Model

- A relation: a set of records (more formally later)
- Schema: a description of data in terms of a data model
  - Name of the schema, name of each field, type of each field

Domain

Schema vs. instance

- Schema (metadata)
  - Specifies how the logical structure of data
  - Is defined at setup time
  - Rarely changes
  - But columns can be added/deleted
- Instance ≡ Snapshot
  - Represents the data content
  - Changes rapidly, but always conforms to the schema

- Schema
  - *User* (*uid* int, *name* string, *age* int, *pop* float)
  - *Group* (*gid* string, *name* string)
  - *Member* (*uid* int, *gid* string)
- Instance
  - *User*: $\{\langle 142, \text{Bart}, 10, 0.9\rangle, \langle 857, \text{Milhouse}, 10, 0.2\rangle, \ldots\}$
  - *Group*: $\{\langle \text{abc}, \text{Book Club}\rangle, \langle \text{gov}, \text{Student Government}\rangle, \ldots\}$
  - *Member*: $\{\langle 142, \text{dps}\rangle, \langle 123, \text{gov}\rangle, \ldots\}$

An instance of the Students relation appears in Figure 3.1. The instance $S1$ contains

FIELDS (ATTRIBUTES, COLUMNS)

Field names

TUPLES (RECORDS, ROWS)

sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

Figure 3.1 An Instance $S1$ of the Students Relation

Basics of Relational Model

Terminology & Concept

The degree (or arity) of a relation is the number of attributes in its relation schema.

Cardinality : Number of rows in any given instance of a relation.

- #rows = *cardinality*
- #fields = *degree / arity*

- Can think of a relation as a *set* of rows or *tuples*.
  – i.e., all rows are distinct

A Table

Table name

Product:

Column names

Name	Price	Category	Manufacturer
gizmo	$19.99	gadgets	GizmoWorks
Power gizmo	$29.99	gadgets	GizmoWorks
SingleTouch	$149.99	photography	Canon
MultiTouch	$203.99	household	Hitachi

Rows

Degree of Product Table = 4 = #Column

Cardinality = 4 = #Row

GATE CSE 2023 | Question: 6

asked Feb 15, 2023 • retagged Dec 22, 2023 by **Hira Thakur**

11,519 views

42

Which one of the options given below refers to the degree (or arity) of a relation in relational database systems?

A. Number of attributes of its relation schema.
B. Number of tuples stored in the relation.
C. Number of entries in the relation.
D. Number of distinct domains of its relation schema.

GATE CSE 2023 | Question: 6

asked Feb 15, 2023 • retagged Dec 22, 2023 by **Hira Thakur**

11,519 views

42

Which one of the options given below refers to the degree (or arity) of a relation in relational database systems?

A. Number of attributes of its relation schema.
B. Number of tuples stored in the relation. Cardinality
C. Number of entries in the relation.
D. Number of distinct domains of its relation schema.

NOTE:

In Relational Model, a relation is a set of rows or tuples.
– i.e., all rows are distinct.

No Two Rows can be exactly same.

In mathematical relations:

- Order of tuples and duplicate tuples do not matter
- Order of attributes is still fixed

NOTE:

A relation is a subset of the Cartesian product of its domains.

R(A B C)

A	B	C
a	1	x
a	1	y
b	1	x

Valid Instance

Domain(A) = {a, b}

Domain(B) = {1}

Domain(C) = {x, y}

$\subseteq$ Dom(A) × Dom(B) × Dom(C)

a	1	x
a	1	y
b	1	x
b	1	y

Note:

$A = \{1, 2, 3\}$

$B = \{a, b\}$

$A \times B$ — Cross product (Cartesian product)

$A \times B = \{(x, y) \mid x \in A, y \in B\}$

$A \times B = \{(1, a), (1, b), (2, a), (2, b), (3, a), (3, b)\}$ — 6 Elements

Tuples and Relationships

- In the *account* relation:
  - Domain of *acct_id* is D_1
  - Domain of *branch_name* is D_2
  - Domain of *balance* is D_3

acct_id	branch_name	balance
A-301	New York	350
A-307	Seattle	275
A-318	Los Angeles	550
...	...	...

account

- The *account* relation is a subset of the tuples in the Cartesian product $D_1 \times D_2 \times D_3$

5.11.1 Relations: ISRO2015-21 top

If $D_1, D_2, \ldots D_n$ are domains in a relational model, then the relation is a table, which is a subset of

A. $D_1 \oplus D_2 \oplus \cdots \oplus D_n$
B. $D_1 \times D_2 \times \cdots \times D_n$
C. $D_1 \cup D_2 \cup \cdots \cup D_n$
D. $D_1 \cap D_2 \cap \cdots \cap D_n$

databases relations isro2015

5.11.1 Relations: ISRO2015-21 top

If $D_1, D_2, \ldots D_n$ are domains in a relational model, then the relation is a table, which is a subset of

A. $D_1 \oplus D_2 \oplus \cdots \oplus D_n$
B. $D_1 \times D_2 \times \cdots \times D_n$
C. $D_1 \cup D_2 \cup \cdots \cup D_n$
D. $D_1 \cap D_2 \cap \cdots \cap D_n$

databases relations isro2015

5.15.2 Relational Calculus: NIELIT 2017 July Scientist B (IT) - Section B: 19 top

If R is a relation in Relational Data Model and $A_1, A_2, \ldots A_n$ are the attributes of relation R, what is the cardinality of R expressed in terms of domain of attributes?

A. $|R| \leq |\text{ dom }(A_1) \times \text{ dom }(A_2) \ldots \text{ dom }(A_n)|$

B. $|R| \geq |\text{ dom } (A_1) \times \text{ dom } (A_2) \ldots \text{ dom } (A_n)|$

C. $|R| = \max(|\text{ dom}(A_1)|, |\text{ dom }(A_2)|, \cdots |\text{ dom }(A_n)|)$

D. $|R| = \min(|\text{ dom}(A_1)|, |\text{ dom }(A_2)|, \cdots |\text{ dom }(A_n)|)$

$R(A_1, A_2, \ldots, A_n)$

Domain D_1

Domain D_n

$R \subseteq D_1 \times D_2 \times \ldots \times D_n$

$|R| \leq |D_1 \times D_2 \times \ldots \times D_n|$

Note:

$X \subseteq Y$

$|X| \leq |Y|$

Relational data model

GO Classes

- A database is a collection of relations (or tables)
- Each relation has a set of attributes (or columns)
- Each attribute has a name and a domain (or type)
  - Values are “atomic”
  - Can be strings, integers, reals, characters,..
  - Cannot be a struct, set, list, array, …
- Each relation contains a set of tuples (or rows)
  - Each tuple has a value for each attribute of the relation
  - Duplicate tuples are not allowed
    - Two tuples are duplicates if they agree on all attributes

☞Recall: Simplicity is a virtue!

Basics of Relational Model

NULL Values

Students (sid, name, passport no.)

101	Sonu	1234
102	monu	NULL

NULL Indicates Value NOT Exist

People (id, name, DOB)

101 Ramkumar Null

Null → Indicates Value Unknown

(Exists but NOT known)

People (name, gender, husband name, wife name)

Sonu F Null Null

Not Exist

Not Applicable

Why Use NULLs

The three meanings of NULL:

- Unknown values
- Inapplicable values
- Empty placeholders

Three meanings of null values

1. not applicable
2. not known
3. absent (not recorded)

Null Values

Attribute values

- are atomic
- have a known domain
- can sometimes be "*null*"

Three meanings of null values

1. not applicable
2. not known
3. absent (not recorded)

Student

studno	name	hons	tutor	year	thesis title
s1	jones	ca	bush	2	*null*
s2	brown	cis	kahn	2	*null*
s3	smith	*null*	goble	2	*null*
s4	bloggs	ca	goble	1	*null*
s5	jones	cs	zobel	1	*null*
s6	peters	ca	kahn	3	*"A CS Survey"*

The NULL Spouse Example

If *employee.spouse* is NULL, does it mean?

- The spouse's name is unknown.
- The employee is not married and therefore has no spouse.

Chapter: The Relational Model

Next Topic:

Keys

The Relational Model

Super Key

Superkey:

a set S of one or more attributes, which, when taken collectively, allows us to identify each tuple uniquely in the table.

i.e. No Two Tuples can have same value of S in the table.

Students (sid, name, passport no.)

a SK

SK

SK

SK

Students (sid, name, passport no.)

101 Sonu

102 Sonu

NOT a SK

Unique Identification of Tuples

102 = sid

SK

102

Sonu

102

C1	C2	C3
1	AB	BC
2	BC	CD
3	CD	AB
1	BD	CD

C_1 : NOT SK

C_2 : SK

C_3 : NOT SK

$\underbrace{C_1 C_3} = \{C_1, C_3\} = SK$

$C_1 C_2 = SK$

$\underbrace{C_1 C_2 C_3} = \{C_1, C_2, C_3\} = SK$

Example of a relation:

Sks :

Year ✓

Gold ✗

Silver ✗

{Gold, Silver} ✗

{Year, Gold} ✓

{Year, Silver} ✓

{Year, Gold, Silver} ✓

Table: MATH_OLYMPIC

4 Sks

Year	Gold	Silver
2008	0	0
2009	0	3
2010	0	2
2011	1	1
2012	2	3
2013	0	2
2014	0	1
2015	0	1
2016	0	1
2017	0	0
2018	0	3

Concept of keys

A subset K of R is a *superkey* of R if, in any legal relation $r(R)$, for all pairs $t1$ and $t2$ of tuples in r such that $t1 \neq t2$, then $t1[K] \neq t2[K]$, i.e. no two tuples in any legal relation $r(R)$ may have the same value on attribute set K.

Basically, a *superkey* is a set of one or more attributes that can uniquely identify an entity in the entity set.

Keys

- Keys are used to distinguish individual tuples
  - A <u>superkey</u> is a set of attributes that uniquely identifies tuples in a relation

acct_id	branch_name	balance
A-301	New York	350
A-307	Seattle	275
A-318	Los Angeles	550
A-319	New York	80
A-322	Los Angeles	275

account

- Example:
  { *acct_id* } is a superkey
- Is { *acct_id*, *balance* } a superkey?
  - Yes! Every tuple has a unique set of values for this combination of attributes.
- Is { *branch_name* } a superkey?
  - No. Each branch can have multiple accounts

R(A B C D)

1	2	–	–
1	3	–	–
2	3	–	–

A : NOT SK

B : NOT SK

AB = {A, B} = SK

NOTE:

1. Set of ALL attributes is Always a Superkey. (Because No Two Tuples can be exactly same in all attributes)

2. If S is SK then every superset of S is also a SK.

The Relational Model

Candidate Key

Candidate key:

CK = a Minimal Superkey

i.e. CK is a SK, But Minimal (No unnecessary attribute)

Students (sid, name, passport no.)

Sk

Sk

ck

NOT a ck

Candidate key : minimal SK

If $S = \{A_1, A_2, --, A_k\}$ is CK then

$S - \{A_i\}$ No longer a SK, for all $1 \le i \le k$.

CK ≡ minimal SK

Basically, a *superkey* is a set of one or more attributes that can uniquely identify an entity in the entity set.

A *candidate key* is a *superkey* for which no proper subset is a superkey, i.e. a minimal *superkey*.

Example of a relation:

Sks:

Year → CK

{Year, Gold} : SK

NOT CKs

{Year, Silver}

{Year, Gold, Silver} : SK

Table: MATH_OLYMPIC

Year	Gold	Silver
2008	0	0
2009	0	3
2010	0	2
2011	1	1
2012	2	3
2013	0	2
2014	0	1
2015	0	1
2016	0	1
2017	0	0
2018	0	3

Concept of keys

1 ck & 4 sks

Table: OSCAR_DIRECTOR

Best Director	Awards	Nominations
John Ford	4	5
William Wyler	3	12
Frank Capra	3	6
Billy Wilder	2	8
David Lean	2	7
Fred Zinnemann	2	7
Steven Spielberg	2	7

In the above relational schema, {Best Director}, {Best Director, Awards}, {Best Director, Nominations} and {Best Director, Awards, Nominations} are all *superkeys* and {Best Director} is the only *candidate key*.

Employee

Emp#	Name	Bdate	Pers#	Dep
1	John	310190	39302	IT
2	Mike	210191	39302	IT
4	Peter	220191	44930	null
5	Eva	230191	88302	Sales
6	Ann	310190	64212	IT
7	John	240191	86921	IT
8	Ann	310190	11267	IT
9	Ann	160792	11267	IT

from this Instance point of view:

Emp# : minimal sk = ck $\longrightarrow$ sk

{Emp#, Name} $\Rightarrow$ sk, but NOT minimal sk $\Rightarrow$ No ck

{Bdate, Pers#} $\Rightarrow$ sk $\Rightarrow$ minimal sk $\Rightarrow$ ck ✓

Superkeys and Candidate Keys

- A superkey is a set of attributes that uniquely identifies tuples in a relation
- Adding attributes to a superkey produces another superkey
  - If { *acct_id* } is a superkey, so is { *acct_id*, *balance* }
  - If a set of attributes $K \subseteq R$ is a superkey, so is any superset of K
  - Not all superkeys are equally useful…
- A *minimal* superkey is called a candidate key
  - A superkey for which no proper subset is a superkey
  - For *account*, only { *acct_id* } is a candidate key

Note:

Sk, Ck : Properties of Schema

→ NOT properties of Instance

NOTE:

Null values are, by default, allowed in the Super Keys & Candidate Keys..

We ignore Null values & SK/CK must uniquely identify Non-Null tuples.

Student (sid, sname, Passport_no)

a ck

a ck with Non-Null Value

1023
Null
Null
1025

fine

a ck

The Relational Model

Primary Key

Primary key: ≡ Captain

One of the candidate keys of a relation is chosen as its primary key, by the database designer.

Primary Key ==> Can’t have Null Values.

- A **superkey** of a relation is a set of one or more attributes whose values are guaranteed to identify tuples in the relation uniquely. A candidate key is a minimal superkey, that is, a set of attributes that forms a superkey, but none of whose subsets is a superkey. One of the candidate keys of a relation is chosen as its **primary key**.

We shall use the term **primary key** to denote a candidate key that is chosen by the database designer as the principal means of identifying tuples within a relation. A key (whether primary, candidate, or super) is a property of the entire relation, rather than of the individual tuples. Any two individual tuples in the relation are prohibited from having the same value on the key attributes at the same time. The designation of a key represents a constraint in the real-world enterprise being modeled.

Students (sid, name, passport no.)

a ck

my primary key

a ck

Alternative keys

The primary key attributes are required to be nonnull and unique;
that is, no tuple can have a null value for a primary-key attribute, and no two tuples in the relation can be equal on all the primary-key attributes.

Eg: R (<u>A B</u> C D)

If AB is a pk then neither A, nor B can have Null value.

Note: Primary key is underlined.

Keys: Definitions

- Superkey
  - a *set* of attributes whose values together *uniquely* identify a tuple in a relation
- Candidate Key
  - a superkey for which no proper subset is a superkey: a superkey that is *minimal*
  - *Can be more than one for a relation*
- Primary Key
  - a candidate key chosen to be the main key
  - *One for each relation*, indicated by underlining the key attributes

Student(<u>studno</u>,name,tutor,year)

Problem 1.(10%) Consider a table $T(A, B, C)$, namely, the table's name is T, and its attributes are A, B, and C. It has 2 candidate keys: $\{A, B\}$ and $\{B, C\}$. Can the following pairs of tuples co-exist in T, respectively?

(i) $(a1, b1, c1)$ and $(a1, b2, c2)$.
(ii) $(a1, b1, c1)$ and $(a1, b2, c1)$.
(iii) $(a1, b1, c1)$ and $(a2, b1, c1)$.

Problem 1.(10%) Consider a table $T(A, B, C)$, namely, the table's name is T, and its attributes are A, B, and C. It has 2 candidate keys: $\{A, B\}$ and $\{B, C\}$. Can the following pairs of tuples co-exist in T, respectively?

(i) $(a1, b1, c1)$ and $(a1, b2, c2)$.
(ii) $(a1, b1, c1)$ and $(a1, b2, c1)$.
(iii) $(a1, b1, c1)$ and $(a2, b1, c1)$.

Answer. (i) yes (ii) yes (iii) no

5.4.1 Candidate Key: UGC NET CSE | January 2017 | Part 3 | Question: 8 top

Which one is correct w.r.t. RDBMS?

A. primary key $\subseteq$ super key $\subseteq$ candidate key
B. primary key $\subseteq$ candidate key $\subseteq$ super key
C. super key $\subseteq$ candidate key $\subseteq$ primary key
D. super key $\subseteq$ primary key $\subseteq$ candidate key

Relational model DBMS

5.4.1 Candidate Key: UGC NET CSE | January 2017 | Part 3 | Question: 8 top

Which one is correct w.r.t. RDBMS?

A. primary key $\subseteq$ super key $\subseteq$ candidate key
B. primary key $\subseteq$ candidate key $\subseteq$ super key
C. super key $\subseteq$ candidate key $\subseteq$ primary key
D. super key $\subseteq$ primary key $\subseteq$ candidate key

ISRO2014-42

3

Let x, y, z, a, b, c be the attributes of an entity set E. If $\{x\}, \{x, y\}, \{a, b\}, \{a, b, c\}, \{x, y, z\}$ are superkeys then which of the following are the candidate keys?

A. $\{x, y\}$ and $\{a, b\}$
B. $\{x\}$ and $\{a, b\}$
C. $\{x, y, z\}$ and $\{a, b, c\}$
D. $\{z\}$ and $\{c\}$

ISRO2014-42

Table E

asked Sep 23, 2015 • edited Jan 24 by **Makhdoom Ghaya**

4,418 views

3

Let x, y, z, a, b, c be the attributes of an entity set E. If $\{x\}, \{x, y\}, \{a, b\}, \{a, b, c\}, \{x, y, z\}$ are superkeys then which of the following are the candidate keys?

A. $\{x, y\}$ and $\{a, b\}$
B. $\{x\}$ and $\{a, b\}$
C. $\{x, y, z\}$ and $\{a, b, c\}$
D. $\{z\}$ and $\{c\}$

SK: $\{x\}$ $\{x, y\}$ $\{x, y, z\}$

minimal sk = ck

NOT ck

NOT minimal sk

$\{a, b\}$, $\{a, b, c\}$

ck

NOT minimal sk

Note: By Default,

$$\boxed{\text{key} \equiv \text{Candidate key}}$$

NOTE:

1. Key $\overset{\text{by default}}{===}$ Candidate Key

2. Every relation must have Exactly One primary key

3. Set of ALL attributes is Always a Super Key.

NOTE:

4. Every Superset of a Candidate key is a Superkey.

5. Candidate keys that are not chosen as the primary key are often termed as alternate keys.

NOTE:

6. Proper Subset of a candidate key is Never a Superkey.

7. No part of the primary key may be null.

Question 1: Which of the following statements are true

1. Each superkey is a superset of some candidate key.

2. Each primary key is also a candidate key, but there may be candidate keys that are not primary keys.

a) only 1 is true

b) only 2 is true

c) both 1 and 2 are true

d) neither 1 nor 2 are true

Question 1: Which of the following statements are true

1. Each superkey is a superset of some candidate key.
2. Each primary key is also a candidate key, but there may be candidate keys that are not primary keys.

a) only 1 is true
b) only 2 is true
c) both 1 and 2 are true
d) neither 1 nor 2 are true

Q :

c. Consider a relation with five attributes (say, A, B, C, D, and E). How many possibilities are there for what its primary key could be (e.g., ACE)?

Q :

c. Consider a relation with five attributes (say, A, B, C, D, and E). How many possibilities are there for what its primary key could be (e.g., ACE)? = 31

R(ABCDE) → Exactly 1 PK but which one?

→ any Nonempty subset could be the PK

→ $2^5 - 1 = 31$ Nonempty subsets

A	AB
B	AC
C	ABC
D	ABDE
E	ABCDE

1 Subtracted for empty subset

!! **Exercise 3.1.3:** Suppose R is a relation with attributes $A_1, A_2, \ldots, A_n$. As a function of n, tell how many superkeys R has, if:

a) The only key is A_1.

b) The only keys are A_1 and A_2.

c) The only keys are $\{A_1, A_2\}$ and $\{A_3, A_4\}$.

d) The only keys are $\{A_1, A_2\}$ and $\{A_1, A_3\}$.

Must See the solution Video if dont understand

Source: Ullman DBMS Book

Video Solution Link in the Pinned Comment.

The Relational Model

Integrity Constraints

INTEGRITY CONSTRAINTS OVER RELATIONS

Constraints == Restriction

Integrity Constraints == Restriction on what data can be stored in the tables.

3.2 INTEGRITY CONSTRAINTS OVER RELATIONS

A database is only as good as the information stored in it, and a DBMS must therefore help prevent the entry of incorrect information. An **integrity constraint (IC)** is a condition that is specified on a database schema, and restricts the data that can be stored in an instance of the database. If a database instance satisfies all the integrity constraints specified on the database schema, it is a **legal** instance. A DBMS **enforces** integrity constraints, in that it permits only legal instances to be stored in the database.

INTEGRITY CONSTRAINTS OVER RELATIONS

Many kinds of integrity constraints can be specified in the relational model.

For example:

No two students have the same sid value. ≡ key IC

Student name can not be Null. ≡ Not Null IC

Age must be an integer. ≡ Domain IC

Age
~~John~~
29

If two tuples have same District name, then those two rows must also have same State name. ≡ functional Dependency IC

GATE Application form ; → Schema

Your Data : Instance

Name* → mandatory field

Name : NOT NULL

IC :

Specified on schema ;

Checked on Runtime

Specified on Schema

Integrity constraints are specified and enforced at different times:

1. When the DBA or end user defines a database schema, he or she specifies the ICs that must hold on any instance of this database.
2. When a database application is run, the DBMS checks for violations and disallows changes to the data that violate the specified ICs.

Integrity constraints are specified and enforced at different times:

1. When the DBA or end user defines a database schema, he or she specifies the ICs that must hold on any instance of this database.

2. When a database application is run, the DBMS checks for violations and disallows changes to the data that violate the specified ICs.

Very Important:

Integrity Constraints:

Referential Integrity Constraint

(Foreign Key Constraint)

3.2.2 Foreign Key Constraints

Sometimes the information stored in a relation is linked to the information stored in another relation. If one of the relations is modified, the other must be checked, and perhaps modified, to keep the data consistent.

4.4.5 Referential Integrity

Often, we wish to ensure that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation. This condition is called **referential integrity**.

Foreign key		
cid	*grade*	*sid*
Carnatic101	C	53831
Reggae203	B	53832
Topology112	A	53650
History105	B	53666
~~DBMS~~	~~D~~	~~6000~~

NOT Allowed

Enrolled (Referencing relation)

Child Relation

Primary key				
sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

Students (Referenced relation)

(Parent relation)

Figure 3.4 Referential Integrity

Q:

Why foreign key is called "foreign" key??

Q:

Why foreign key is called "foreign" key??

Because foreign key is a Primary Key in a foreign table.

NOTE:

a foreign key FK of a table R is a set of attributes of R that *references* the primary key of a foreign table.

So, FK of R is "key in foreign table". Hence, it is called Foreign Key.

Figure 5.6
One possible database state for the COMPANY relational database schema.

NOT a SK

FK

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1
Sonu	-	-	-	-	-	-	-	-	10

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire

Figure 5.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

a fk

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire

Note:

R (A B C) S (X Y)

Primary key

R(C) depends on S(X) ≡

R(C) is a FK which Refers to primary key of S

Note:

Refers to

R (A B C) S (<u>X</u> Y)

Primary key

FK

R: Referencing Relation (Child Table)

S: Referenced Relation (Parent Table)

1. FK Refers to PK.
2. FK itself may not be a PK/CK/SK.
3. Every Non-Null value that apperas in FK must be in Referenced attribute.

The **referential integrity constraint** is specified between two relations and is used to maintain the consistency among tuples in the two relations. Informally, the referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an *existing tuple* in that relation. For example, in Figure 5.6, the attribute Dno of EMPLOYEE gives the department number for which each employee works; hence, its value in every EMPLOYEE tuple must match the Dnumber value of some tuple in the DEPARTMENT relation.

Foreign Key Constraint:

Referring to the same table: Possible.

Students (Sid, Sname, supervisor_sid)

primary key

FK

Sid	Sname	supervisor_sid
101	monu	102
102	sonu	Null
103	Chintu	102
104	John	101

Allowed

Note: R(A B C D)

A: PK

D: FK

① FK always refers to PK.

② Null values are, by default, allowed in FK.

③ D → A

D: FK, A: PK

D: Referencing attribute

A: Referenced attribute

Note: R(A B C D)

PK — A

FK — D

(4) Referential Integrity Constraint says

"Every Non-Null value of FK (D) must be in Referenced attribute (A).

Figure 5.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE PK Fk

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire

Foreign Key Constraint:

Many FKs in same table: Yes, possible.

Figure 5.6
One possible database state for the COMPANY relational database schema.

Fk1 Fk2

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

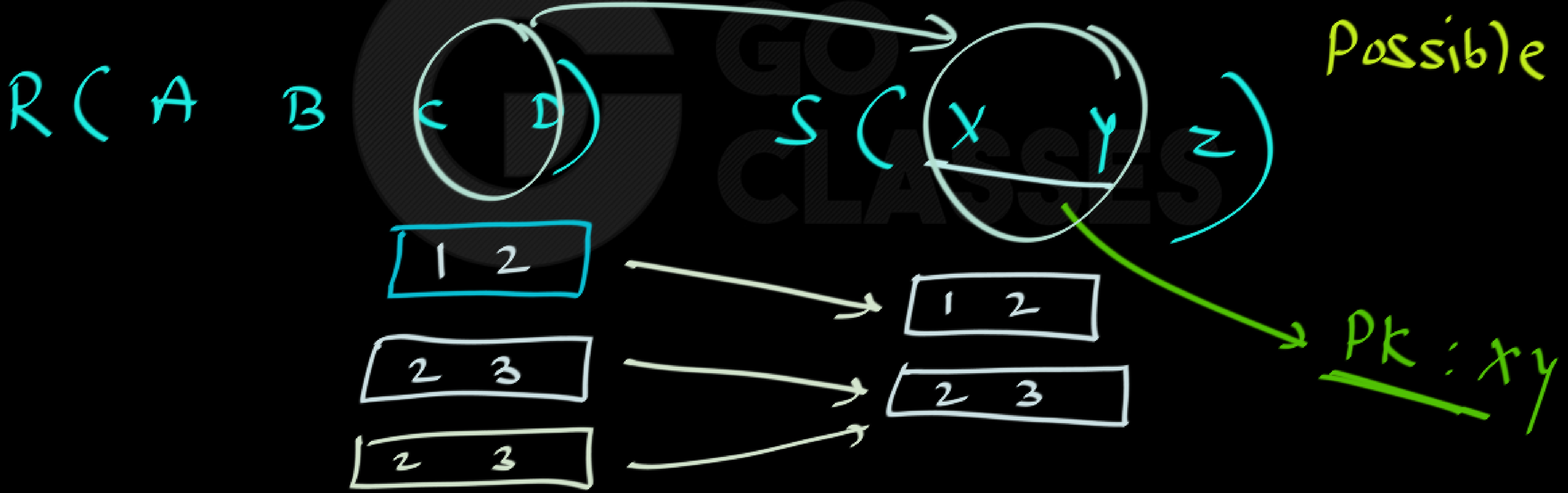
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire

Foreign Key Constraint:

FK containing multiple attributes: Yes, Possible.



In SQL we may declare an attribute or attributes of one relation to be a *foreign key*, referencing some attribute(s) of a second relation (possibly the same relation). The implication of this declaration is twofold:

1. The referenced attribute(s) of the second relation must be declared `UNIQUE` or the `PRIMARY KEY` for their relation. Otherwise, we cannot make the foreign-key declaration.

2. Values of the foreign key appearing in the first relation must also appear in the referenced attributes of some tuple.

Referential Integrity

- Ensures that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

Referential Integrity Constraint (Foreign Key Constraint)

Implementation using SQL

(In the schema of Referencing Table)

NOT Imp for GATE

Foreign key (Enrolled: sid) → Primary key (Students: sid)

cid	*grade*	*sid*
Carnatic101	C	53831
Reggae203	B	53832
Topology112	A	53650
History105	B	53666

Enrolled (Referencing relation)

sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

Students (Referenced relation)

Figure 3.4 Referential Integrity

Since "Enrolled" Table has the FK :

Specifying Foreign Key Constraints in SQL-92

Let us define Enrolled(*sid:* `string`, *cid:* `string`, *grade:* `string`):

Refers to PK of Students

```
CREATE TABLE Enrolled ( sid    CHAR(20),
                        cid    CHAR(20),
                        grade  CHAR(10),
                        PRIMARY KEY (sid, cid),
                        FOREIGN KEY (sid) REFERENCES Students )
```

The foreign key constraint states that every *sid* value in Enrolled must also appear in Students, that is, *sid* in Enrolled is a foreign key referencing Students. Incidentally,

Foreign Keys

A set of attributes in a relation that exactly matches the primary key in another relation

– the names of the attributes don't have to be the same but must be of the same domain

Student (studno, name, hons, tutor, tutorroom, year)

Staff (lecturer, roomno, appraiser, approom)

Notation:

FK1: Student (tutor, tutorroom) references Staff (lecturer, roomno)

FK2: Staff (appraiser, approom) references Staff (lecturer, roomno)

- **foreign key** $(A_{k_1}, A_{k_2}, \ldots, A_{k_n})$ **references** s: The **foreign key** specification says that the values of attributes $(A_{k_1}, A_{k_2}, \ldots, A_{k_n})$ for any tuple in the relation must correspond to values of the primary key attributes of some tuple in relation s.

Referential Integrity Constraint (Foreign Key Constraint)

Actions Taken by DBMS to Enforce Ref. Int. Constraint:

from any FK Point of view:

1. Referencing Table (Child Table)
2. Referenced Table (Parent Table)

Same or Diff

Referential Integrity Constraint (Foreign Key Constraint)

Actions Taken by DBMS to Enforce Ref.

Int. Constraint:

- When Referenced Relation is updated.
- When Referencing Relation is updated.

When Referenced Relation is updated

Parent Table

Actions Taken by DBMS to Enforce Ref. Int. Constraint:

When Referenced Relation is updated:

If NO Violation happens then simply carry out the operation.

Actions Taken by DBMS to Enforce Ref. Int. Constraint:

When Referenced Relation is updated:

Insertion in Parent Table NEVER causes Referential Integrity Violation.

Consider the following tables $T1$ and $T2$.

$T_1(Q)$: FK

Refers to

PK R on T_2

FK

PK

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

$T2$

R	S
2	2
8	3
3	2
9	7
5	7
7	2
10	3

Referenced Relation

Actions Taken by DBMS to Enforce Ref. Int. Constraint:

When Referenced Relation is updated:

IF Violation happens then Three Choices:

1. No Action : Reject the op^n
2. Cascade : Ripple effect
3. Set Null or set Default

NO ACTION

1. *The Default Policy: Reject Violating Modifications.* SQL has a default policy that any modification violating the referential integrity constraint is rejected.

Consider the following tables $T1$ and $T2$.

Delete (7,2) from Parent Table :

No Violation of Ref.Int.

So; Just do it.

FK

PK

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

$T2$

R	S
2	2
8	3
3	2
9	7
5	7
~~7~~	~~2~~

Referenced Table

So; Just Delete (7,2) from T_2.

Consider the following tables $T1$ and $T2$.

Delete (8,3) from Referenced Table T_2

Fk

Pk

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

$T2$

R	S
2	2
~~8~~	~~3~~
3	2
9	7
5	7
7	2

Violation of Ref. Int. Happens.

Policy : No Action ⇒ Deletion NOT Happen

Since Violation happens because of Deletion of (8,3) from T_2 ; So; We Reject this Deletion opn.

Consider the following tables $T1$ and $T2$.

Delete (8,3) from Referenced Table T_2

Violation of Ref. Int. Happens.

Policy : On Delete Cascade

T1

P	Q
2	2
~~3~~	~~8~~
7	3
~~5~~	~~8~~
6	9
8	5
~~9~~	~~8~~

FK

T2

R	S
2	2
~~8~~	~~3~~
3	2
9	7
5	7
7	2

PK

Ripple Effect of Deletion

(3,8), (5,8) (9,8) will Also be Deleted from T_1

Consider the following tables $T1$ and $T2$.

Delete (8,3) from Referenced Table T_2

FK

PK

Violation of Ref. Int. Happens.

Policy: On Delete Set Null

T1

P	Q
2	2
3	~~8~~ Null
7	3
5	~~8~~ Null
6	9
8	5
9	~~8~~ Null

T2

R	S
2	2
~~8~~	~~3~~
3	2
9	7
5	7
7	2

Q attribute becomes Null in Rows where Violation occurs.

Consider the following tables $T1$ and $T2$.

Delete (8,3) from Referenced Table T_2

Violation of Ref. Int. Happens.

Policy: on Delete set 7

FK

PK

$T1$

P	Q
2	2
3	~~8~~ 7
7	3
5	~~8~~ 7
6	9
8	5
9	~~8~~ 7

$T2$

R	S
2	2
~~8~~	~~3~~
3	2
9	7
5	7
7	2

Consider the following tables $T1$ and $T2$.

Delete (8,3) from T_2

Violation of Ref. Int. happens

Policy: on Delete set 5

FK

$T1$

P	Q
2	2
3	~~8~~ 5
7	3
5	~~8~~ 5
6	9
8	5
9	~~8~~ 5

$T2$

R	S
2	2
~~8~~	~~3~~
3	2
9	7
5	7
7	2

PK

Referenced relation

CASCADE [edit]

Whenever rows in the parent (referenced) table are deleted (or updated), the respective rows of the child (referencing) table with a matching foreign key column will be deleted (or updated) as well. This is called a cascade delete (or update).

2. *The Cascade Policy.* Under this policy, changes to the referenced attribute(s) are mimicked at the foreign key. For example, under the cas-

Consider the following tables $T1$ and $T2$.

Update T_2 → Referenced Table

(7,2) → (10,3)

No Violation of Ref. Int.

→ Just Do it.

FK

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

Pk

$T2$

R	S
2	2
8	3
3	2
9	7
5	7
~~7~~	~~2~~
10	3

Referenced Relation

Done

Consider the following tables $T1$ and $T2$.

update T_2 : Parent Table

$(5,7) \rightarrow (1,2)$

Violation Happens

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

Fk

$T2$

R	S
2	2
8	3
3	2
9	7
~~5~~	~~7~~
7	2

Pk

Referenced Relation

Policy 1: on update No Action

→ Reject Update opn of T_2 (DBms Rejects our T_2 update)

Consider the following tables $T1$ and $T2$.

update T_2 : Parent Table

$(5,7) \longrightarrow (1,2)$

Violation Happens

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	~~5~~ 1
9	8

Fk

$T2$

R	S
2	2
8	3
3	2
9	7
~~5~~	~~7~~
7	2

Pk

Referenced Relation

Policy 2 : on update Cascade

Ripple effect of updation

$\Rightarrow$ So; $T_1(8,5)$ become $T_1(8,1)$

Consider the following tables $T1$ and $T2$.

update T_2 : Parent Table

$(5,7) \longrightarrow (1,2)$

Violation Happens

T1

P	Q
2	2
3	8
7	3
5	8
6	9
8	~~5~~ Null
9	8

T2

R	S
2	2
8	3
3	2
9	7
~~5~~	~~7~~
7	2

Fk

Pk

Referenced Relation

Policy : On update set Null

So, $T_1(8,5)$ becomes $T_1(8, Null)$

Consider the following tables $T1$ and $T2$.

update T_2 : Parent Table

$(5,7) \longrightarrow (1,2)$

Violation Happens

Policy : On update set 7

So, $T_1(8,5)$ becomes $T_1(8,7)$

FK

PK

Referenced Relation

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	~~5~~ 7
9	8

$T2$

R	S
2	2
8	3
3	2
9	7
~~5~~	~~7~~
7	2

SET NULL, SET DEFAULT [edit]

In general, the action taken by the DBMS for SET NULL or SET DEFAULT is the same for both ON DELETE or ON UPDATE: the value of the affected referencing attributes is changed to NULL for SET NULL, and to the specified default value for SET DEFAULT.

3. *The Set-Null Policy.* Here, when a modification to the referenced relation affects a foreign-key value, the latter is changed to `NULL`. For instance, if

SQL Implementation of Policies:

Action

These Actions are specified in the Referencing Table

Note: FK, Actions $\xrightarrow{\text{Specified in}}$ Referencing Table

SQL Implementation of Actions:

```
CREATE TABLE Enrolled ( sid      CHAR(20),
                        cid      CHAR(20),
                        grade    CHAR(10),
                        PRIMARY KEY (sid, cid),
                        FOREIGN KEY (sid) REFERENCES Students
                            ON DELETE CASCADE
                            ON UPDATE NO ACTION )
```

Referencing Table

These options may be chosen for deletes and updates, independently, and they are stated with the declaration of the foreign key. We declare them with ON DELETE or ON UPDATE followed by our choice of SET NULL or CASCADE.

```
CREATE TABLE Enrolled ( sid    CHAR(20),
                        cid    CHAR(20),
                        grade  CHAR(10),
                        PRIMARY KEY (sid, cid),
                        FOREIGN KEY (sid) REFERENCES Students
                                    ON DELETE CASCADE
                                    ON UPDATE NO ACTION )
```

When Referencing Relation is updated

Child relation

Actions Taken by DBMS to Enforce Ref. Int. Constraint:

When Referencing Relation is updated:

If NO Violation happens then Simply Carry Out the operation.

Actions Taken by DBMS to Enforce Ref. Int. Constraint:

When Referencing Relation is updated:

Deletion in Child Table NEVER causes Referential Integrity Violation.

Consider the following tables $T1$ and $T2$.

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

$T2$

R	S
2	2
8	3
3	2
9	7
5	7
7	2

Actions Taken by DBMS to Enforce Ref. Int. Constraint:

When Referencing Relation is updated:

IF Violation happens then:

the system has to reject the violating modification.

3.11.1 Referential Integrity: GATE CSE 1997 | Question: 6.10, ISRO2016-54 top

https://gateoverflow.in/2266

Let $R(a, b, c)$ and $S(d, e, f)$ be two relations in which d is the foreign key of S that refers to the primary key of R. Consider the following four operations R and S

I. Insert into R

II. Insert into S

III. Delete from R

IV. Delete from S

Which of the following can cause violation of the referential integrity constraint above?

A. Both I and IV

B. Both II and III

C. All of these

D. None of these

UGC NET CSE | July 2018 | Part 2 | Question: 69

will ≡ 100%

Let $R_1(a, b, c)$ and $R_2(x, y, z)$ be two relations in which a is the foreign key of R_1 that refers to the primary key of R_2. Consider the following four options.

a. Insert into R_1
b. Insert into R_2 — Never
c. Delete from R_1 — Never
d. Delete from R_2

R_1 : Child Table

R_2 : Parent Table

Which of the following is correct about the referential integrity constraint with respect to above?

A. Operations a and b will cause violation
B. Operations b and c will cause violation
C. Operations c and d will cause violation
D. Operations d and a will cause violation

a, d : may cause violation

3.11.1 Referential Integrity: GATE CSE 1997 | Question: 6.10, ISRO2016-54 top

https://gateoverflow.in/2266

Let $R(a, b, c)$ and $S(d, e, f)$ be two relations in which d is the foreign key of S that refers to the primary key of R. Consider the following four operations R and S

R(a, b, c) S(d, e, f)

FK

Assume a

I. $\boxed{\text{Insert into } R}$: NEVER Cause Ref. Int. Violation

II. Insert into S

III. Delete from R

IV. Delete from S → Never

Which of the following can cause violation of the referential integrity constraint above?

A. Both I and IV

B. Both II and III ✓

C. All of these

D. None of these

R : Referenced Relation (Parent)

S : Referencing Relation (Child)

Consider the following tables $T1$ and $T2$.

Delete (7,2) from Parent Table

No Violation

FK

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

Parent

$T2$

R	S
2	2
8	3
3	2
9	7
5	7
~~7~~	~~2~~

5.11.4 Referential Integrity: GATE CSE 2021 Set 2 | Question: 6

Consider the following statements $S1$ and $S2$ about the relational data model:

- $S1$: A relation scheme can have at most one foreign key.
- $S2$: A foreign key in a relation scheme R cannot be used to refer to tuples of R.

Which one of the following choices is correct?

A. Both $S1$ and $S2$ are true
B. $S1$ is true and $S2$ is false
C. $S1$ is false and $S2$ is true
D. Both $S1$ and $S2$ are false

gatecse-2021-set2 databases referential-integrity 1-mark

Correct Version: a Relation can have 0 or more FKs.

5.11.4 Referential Integrity: GATE CSE 2021 Set 2 | Question: 6

Consider the following statements $S1$ and $S2$ about the relational data model:

- $S1$: A relation scheme can have at most one foreign key. ✗
- $S2$: A foreign key in a relation scheme R cannot be used to refer to tuples of R. ✗

Which one of the following choices is correct?

A. Both $S1$ and $S2$ are true

B. $S1$ is true and $S2$ is false

C. $S1$ is false and $S2$ is true

D. Both $S1$ and $S2$ are false ✓

gatecse-2021-set2 databases referential-integrity 1-mark

Figure 5.6
One possible database state for the COMPANY relational database schema.

Fk2 Fk1

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire

5.11.3 Referential Integrity: GATE CSE 2017 Set 2 | Question: 19

Consider the following tables $T1$ and $T2$.

$T1$

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

$T2$

R	S
2	2
8	~~3~~ 2
3	2
9	7
5	7
7	2

In table $T1$ **P** is the primary key and **Q** is the foreign key referencing **R** in table $T2$ with on-delete cascade and on-update cascade. In table $T2$, **R** is the primary key and **S** is the foreign key referencing **P** in table $T1$ with on-delete set NULL and on-update cascade. In order to delete record $\langle 3, 8 \rangle$ from the table $T1$, the number of additional records that need to be deleted from table $T1$ is ________

5.11.3 Referential Integrity: GATE CSE 2017 Set 2 | Question: 19

Consider the following tables $T1$ and $T2$.

Q : FK$_1$

on Delete Cascade
on update Cascade

Child Table : T_1
Parent Table : T_2

$T1$

P	Q
2	2
~~3~~	~~8~~
7	3
5	8
6	9
8	5
9	8

$T2$

R	S
2	2
8	3
3	2
9	7
5	7
7	2

S : FK$_2$

on Delete set Null
on update cascade

Child Table : T_2
Parent Table : T_1

In table $T1$ **P** is the primary key and **Q** is the foreign key referencing **R** in table $T2$ with on-delete cascade and on-update cascade. In table $T2$, **R** is the primary key and **S** is the foreign key referencing **P** in table $T1$ with on-delete set NULL and on-update cascade. In order to delete record $\langle 3, 8 \rangle$ from the table $T1$, the number of additional records that need to be deleted from table $T1$ is ________

5.11.3 Referential Integrity: GATE CSE 2017 Set 2 | Question: 19

Consider the following tables $T1$ and $T2$.

FK1

FK2

Fk1 has No violation

for Fk2 = R

we are Deleting from Referenced Table

$T1$

P	Q
2	2
~~3~~	~~8~~
7	3
5	8
6	9
8	5
9	8

$T2$

R	S
2	2
8	3
3	2
9	7
5	7
7	2

In table $T1$ **P** is the primary key and **Q** is the foreign key referencing **R** in table $T2$ with on-delete cascade and on-update cascade. In table $T2$, **R** is the primary key and **S** is the foreign key referencing **P** in table $T1$ with on-delete set NULL and on-update cascade. In order to delete record $\langle 3, 8 \rangle$ from the table $T1$, the number of additional records that need to be deleted from table $T1$ is ________

R (A B C)   S (D $\underline{E}$ F G)

FK

for FK = C ; Child Table = R

If we Delete from R ; then this FK = C has No problem.

5.11.3 Referential Integrity: GATE CSE 2017 Set 2 | Question: 19

Consider the following tables $T1$ and $T2$.

Parent Table

w.r.t. $fk_2 = S$

$T1$

P	Q
2	2
~~3~~	~~8~~
7	3
5	8
6	9
8	5
9	8

$T2$

R	S
2	2
8	3
3	2
9	7
5	7
7	2

Null

$fk_2 = S$

T_1: Parent Table

$\Rightarrow$ on Delete set Null

[redacted]

Violation

In table $T1$ **P** is the primary key and **Q** is the foreign key referencing **R** in table $T2$ with on-delete cascade and on-update cascade. In table $T2$, **R** is the primary key and **S** is the foreign key referencing **P** in table $T1$ with on-delete set NULL and on-update cascade. In order to delete record $\langle 3, 8 \rangle$ from the table $T1$, the number of additional records that need to be deleted from table $T1$ is 0

No additional Deleted.

5

Following table has two attributes Employee_id and Manager_id, where Employee_id is a primary key and manager_id is a foreign key referencing Employee_id with on-delete cascade:

	Employee_id	Manager_id
t_1	20	40
t_2	25	40
t_3	30	35
t_4	35	20
t_5	40	45
t_6	45	25

$t_4 \rightarrow t_1$

$t_3 \rightarrow t_4$

No one $\rightarrow t_3$

On deleting the table $(20, 40)$, the set of other tuples that must be deleted to maintain the referential integrity of table is

A. $(30, 35)$ only
B. $(30, 35)$ and $(35, 20)$ only
C. $(35, 20)$ only $\longrightarrow$ imp
D. $(40, 45)$ and $(25, 40)$ only

UGC NET CSE | June 2019 | Part 2 | Question: 39

5

Following table has two attributes Employee_id and Manager_id, where Employee_id is a primary key and manager_id is a foreign key referencing Employee_id with on-delete cascade:

	Employee_id	Manager_id
t_1	20	40
t_2	25	40
t_3	30	35
t_4	35	20
t_5	40	45
t_6	45	25

~~t_4~~ $\rightarrow$ ~~t_1~~

~~t_3~~ $\rightarrow$ ~~t_4~~

No one $\rightarrow$ t_3

On deleting the table $(20, 40)$, the set of other tuples that must be deleted to maintain the referential integrity of table is

A. $(30, 35)$ only

B. $(30, 35)$ and $(35, 20)$ only ✓

C. $(35, 20)$ only

D. $(40, 45)$ and $(25, 40)$ only

UGC NET CSE | June 2019 | Part 2 | Question: 39

5

Following table has two attributes Employee_id and Manager_id, where Employee_id is a primary key and manager_id is a foreign key referencing Employee_id with on-delete cascade:

	Employee_id	Manager_id
t_1	20	40
t_2	25	40
t_3	30	35
t_4	35	20
t_5	40	45
t_6	45	25

By deleting t_1

t_4 is affected

On deleting the table $(20, 40)$, the set of other tuples that must be deleted to maintain the referential integrity of table is

A. $(30, 35)$ only
B. $(30, 35)$ and $(35, 20)$ only
C. $(35, 20)$ only
D. $(40, 45)$ and $(25, 40)$ only

UGC NET CSE | June 2019 | Part 2 | Question: 39

5

Following table has two attributes Employee_id and Manager_id, where Employee_id is a primary key and manager_id is a foreign key referencing Employee_id with on-delete cascade:

By deleting t_4

t_3 affected

Employee_id	Manager_id
~~20~~	~~40~~
25	40
30	35
~~35~~	~~20~~
40	45
45	25

t_4:

FK : m_id

~~on Delete cascade~~

On deleting the table $(20, 40)$, the set of other tuples that must be deleted to maintain the referential integrity of table is

A. $(30, 35)$ only
B. $(30, 35)$ and $(35, 20)$ only
C. $(35, 20)$ only
D. $(40, 45)$ and $(25, 40)$ only

UGC NET CSE | June 2019 | Part 2 | Question: 39

5

Following table has two attributes Employee_id and Manager_id, where Employee_id is a primary key and manager_id is a foreign key referencing Employee_id with on-delete cascade:

by t_3 Deletion

No one Affected

Employee_id	Manager_id
20	40
25	40
30	35
35	20
40	45
45	25

Fk : m-id

on Delete cascade

On deleting the table $(20, 40)$, the set of other tuples that must be deleted to maintain the referential integrity of table is

A. $(30, 35)$ only
B. $(30, 35)$ and $(35, 20)$ only
C. $(35, 20)$ only
D. $(40, 45)$ and $(25, 40)$ only

UGC NET CSE | June 2019 | Part 2 | Question: 39

5.32.2 Referential Integrity: UGC NET CSE | September 2013 | Part 3 | Question: 24 top

_____ constraints ensure that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

A. Logical Integrity
B. Referential Integrity
C. Domain Integrity
D. Data Integrity

ugcnetcse-sep2013-paper3 databases referential-integrity

5.32.2 Referential Integrity: UGC NET CSE | September 2013 | Part 3 | Question: 24 top

____ constraints ensure that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

A. Logical Integrity
B. Referential Integrity
C. Domain Integrity
D. Data Integrity

ugcnetcse-sep2013-paper3 databases referential-integrity

5.13.1 Referential Integrity: NIELIT 2018-49 top

The following table has two attributes X and Y where X is the primary key and Y is the foreign key referencing X with on-delete cascade.

X	Y
2	4
3	4
4	3
5	2
7	2
9	5
6	4

The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple $(3,4)$ is deleted is

A. $(4,3)$ and $(6,4)$ B. $(2,4)$ and $(7,2)$ C. $(3,2)$ and $(9,5)$ D. $(3,4),(4,5)$ and $(6,4)$

5.13.1 Referential Integrity: NIELIT 2018-49 top

The following table has two attributes X and Y where X is the primary key and Y is the foreign key referencing X with on-delete cascade.

	X	Y	
t_1	2	4	
t_2	3	4	
t_3	4	3	
	5	2	t_4
	7	2	t_5
	9	5	t_6
	6	4	t_7

$Y \to$ fk

$t_3 \to t_2$

$t_7, t_1, t_2 \to t_3$

$t_4, t_5 \to t_1$

Noone $\to t_7$

Noone $\to t_5$

$t_6 \to t_4$

Noone $\to t_6$

The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple $(3,4)$ is deleted is

A. $(4,3)$ and $(6,4)$ B. $(2,4)$ and $(7,2)$ C. $(3,2)$ and $(9,5)$ D. $(3,4),(4,5)$ and $(6,4)$

5.13.1 Referential Integrity: NIELIT 2018-49 top

The following table has two attributes X and Y where X is the primary key and Y is the foreign key referencing X with on-delete cascade.

	X	Y (fk)
t_1	2	4
t_2	3	4
t_3	4	3
t_4	5	2
t_5	7	2
t_6	9	5
t_7	6	4

~~t3~~ → ~~t2~~

~~t7, t1, t2~~ → ~~t3~~

~~t4, t5~~ → ~~t1~~

Noone → t7

Noone → t5

~~t6~~ → ~~t4~~

Noone → t6

Ans: None

All Tuples gone

The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple $(3,4)$ is deleted is

A. $(4,3)$ and $(6,4)$ B. $(2,4)$ and $(7,2)$ C. $(3,2)$ and $(9,5)$ D. $(3,4),(4,5)$ and $(6,4)$

5.13.1 Referential Integrity: NIELIT 2018-49 top

The following table has two attributes X and Y where X is the primary key and Y is the foreign key referencing X with on-delete cascade.

X	Y
2	4
3	4
4	3
5	2
7	2
9	5
6	4

All Tuples Deletes ✓

Ans: No option matches ✓

The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple $(3,4)$ is deleted is

A. $(4,3)$ and $(6,4)$ ✗
B. $(2,4)$ and $(7,2)$ ✗
C. $(3,2)$ and $(9,5)$ ✗
D. $(3,4),(4,5)$ and $(6,4)$ ✗

3.11.2 Referential Integrity: GATE CSE 2005 | Question: 76 top

https://gateoverflow.in/1399

The following table has two attributes A and C where A is the primary key and C is the foreign key referencing A with on-delete cascade.

A	C
2	4
3	4
4	3
5	2
7	2
9	5
6	4

The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple $(2, 4)$ is deleted is:

A. $(3, 4)$ and $(6, 4)$
B. $(5, 2)$ and $(7, 2)$
C. $(5, 2), (7, 2)$ and $(9, 5)$
D. $(3, 4), (4, 3)$ and $(6, 4)$

Video Solution Link in the Pinned Comment.