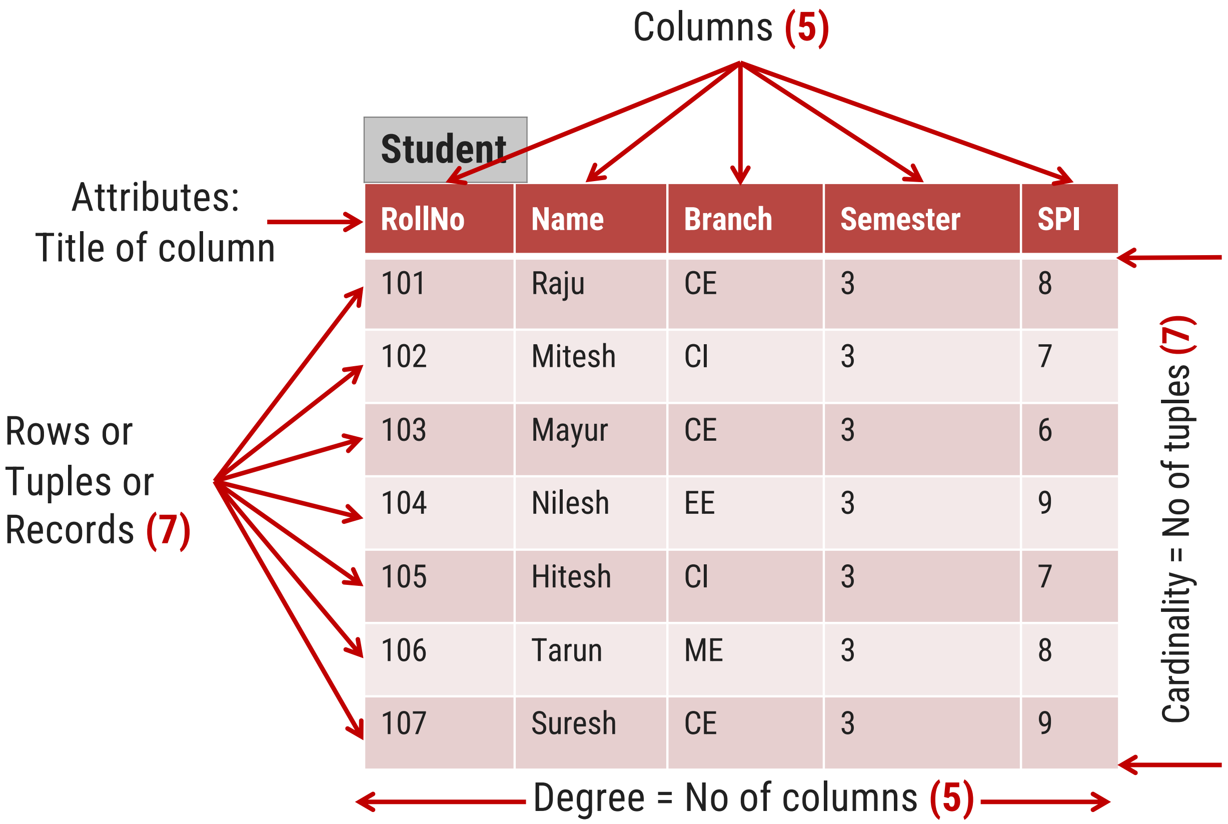
Integrity Constraints

Introduction to relational model

- Relational Model was proposed by E.F. Codd to model data in the form of relations or tables.
- The relational model is the theoretical basis of relational databases.
- Relational Model represents how data is stored in Relational Databases.

Structure of Relational Databases



RollNo	Name	Branch	Semester	SPI
101	Raju	CE	3	8
102	Mitesh	CI	3	7
103	Mayur	CE	3	6
104	Nilesh	EE	3	9
105	Hitesh	CI	3	7
106	Tarun	ME	3	8
107	Suresh	CE	3	9

Table (Relation): A database object that holds a collection of data for a specific topic.
Table consist of rows and columns.

Column (Attribute): The vertical component of a table. A column has a name or attribute.

Record (Tuple): The horizontal component of a table, consisting of a sequence of values, one for each column of the table. It is also known as row.

A database consists of a collection of tables (relations), each having a unique name.

Domain is a set of **all allowable values** for a specific column.
Domain of Branch attribute is (CE, CI, ME, EE)

Introduction to relational model

- Relation Schema: A relation schema represents name of the relation with its attributes. e.g.; STUDENT (ROLL_NO, NAME, branch, semester, SPI) is relation schema for STUDENT.
- If a schema has more than 1 relation, it is called Relational Schema.
- Relation Instance: The set of tuples of a relation at a particular instance of time is called as relation instance. Table 1 shows the relation instance of STUDENT at a particular time. It can change whenever there is insertion, deletion or updation in the database.

- Degree: The number of columns or attributes in the relation is known as degree of the relation. The STUDENT relation defined above has degree 5.

- Cardinality: The number of rows or tuples in a relation is known as cardinality. The STUDENT relation defined above has cardinality 7.
- Relation key: Every row has one or multiple attributes, that can uniquely identify the row in the relation, which is called relation key or primary key.

Key

Types of keys

Apart form uniquely identifying any record, key is also used to establish relationships between tables.

- Super key
- Candidate key
- Primary key
- Alternate key
- Foreign key
- Composite key

Super Key

- A super key is a set of one or more **attributes whose values uniquely identifies each record** within a relation (table).
- A table can have many super keys.
- A super key may have additional attribute that are not needed for unique identity.

Name	Roll No.	Branch
Ajay	101	ENC
Tanya	102	ENC
Rohan	104	CSE
Ajay	106	CSE
Jatin	108	EC

Student Name and Roll No. is a super key.

Super Key
EnrollNo

Super Key
(RollNo, Branch, Sem)

~~Super Key
(SPI, Name, BL)~~

EnrollNo	RollNo	Branch	Sem	SPI	Name	BL
190540107001	101	CE	3	8	Raju	0
190540107002	102	CE	3	7	Mitesh	1
190540106001	101	CI	3	6	Mayur	2
190540106002	102	CI	3	9	Nilesh	0
180540107001	101	CE	5	7	Hitesh	1
180540106001	101	CI	5	8	Raju	0
180540106002	102	CI	5	9	Suresh	0

Candidate Key

- A candidate key is a **subset of a super key**.
- A candidate key is a single attribute or the least combination of attributes that uniquely identifies each record in the table.
- A candidate key is a super key whose proper subset cannot be a key.
- Every candidate key is a super key but every super key is not a candidate key.
- A candidate key is a minimal super key, or a super key with no redundant attributes.
- Candidate keys are not allowed to have NULL values.

Name	Roll No.	Branch
Ajay	101	ENC
Tanya	102	ENC
Rohan	104	CSE
Ajay	106	CSE
Jatin	108	EC

{Roll No.}, {Name, Branch}

Candidate Key
EnrollNo

Candidate Key
(RollNo, Branch, Sem)

EnrollNo	RollNo	Branch	Sem	SPI	Name	BL
190540107001	101	CE	3	8	Raju	0
180540107002	101	CE	5	7	Mitesh	1
190540106001	101	CI	3	6	Mayur	2
180540106002	101	CI	5	9	Nilesh	0

Primary Key

- A primary key is a **candidate key that is chosen by database designer** to identify tuples uniquely in a relation (table) or manipulating the database.

Primary Key

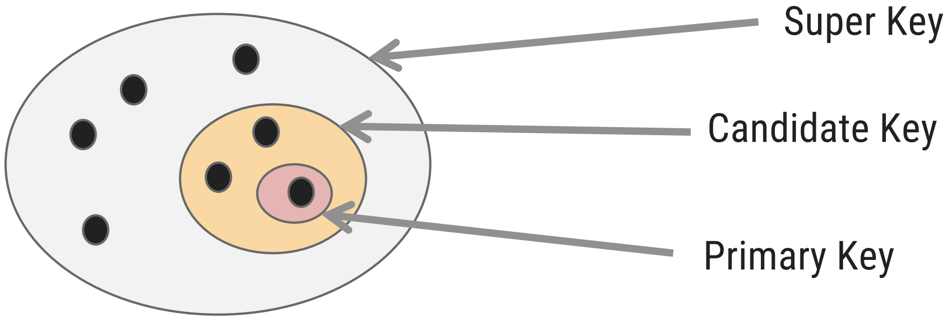
Candidate Key
EnrollNo

Candidate Key
(RollNo, Branch, Sem)

EnrollNo	RollNo	Branch	Sem	SPI	Name	BL
190540107001	101	CE	3	8	Raju	0
180540107002	101	CE	5	7	Mitesh	1
190540106001	101	CI	3	6	Mayur	2
180540106002	101	CI	5	9	Nilesh	0

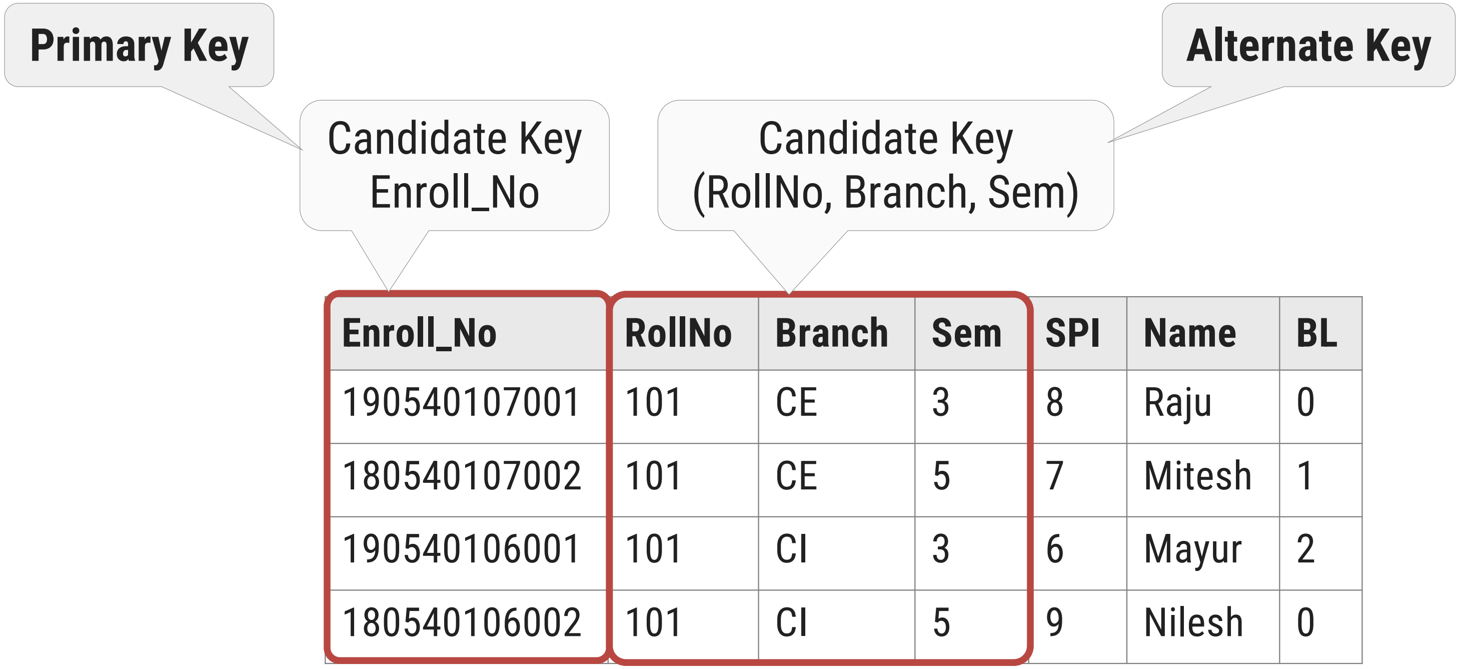
Primary Key rules

- A primary key **may have one or more attributes**.
- There is **only one primary key** in the relation (table).
- The value of primary key **can never be NULL**.
- Generally, the **value of a primary key attribute does not change**.
- The value of primary key must always be unique.
- A relation is allowed to have only one primary key.



Alternate Key

- An alternate key is a **candidate key that is not chosen by database designer** to identify tuples uniquely in a relation.
- Out of all candidate keys, only one gets selected as primary key, remaining keys are known as alternate keys.



Primary Key

Candidate Key
Enroll_No

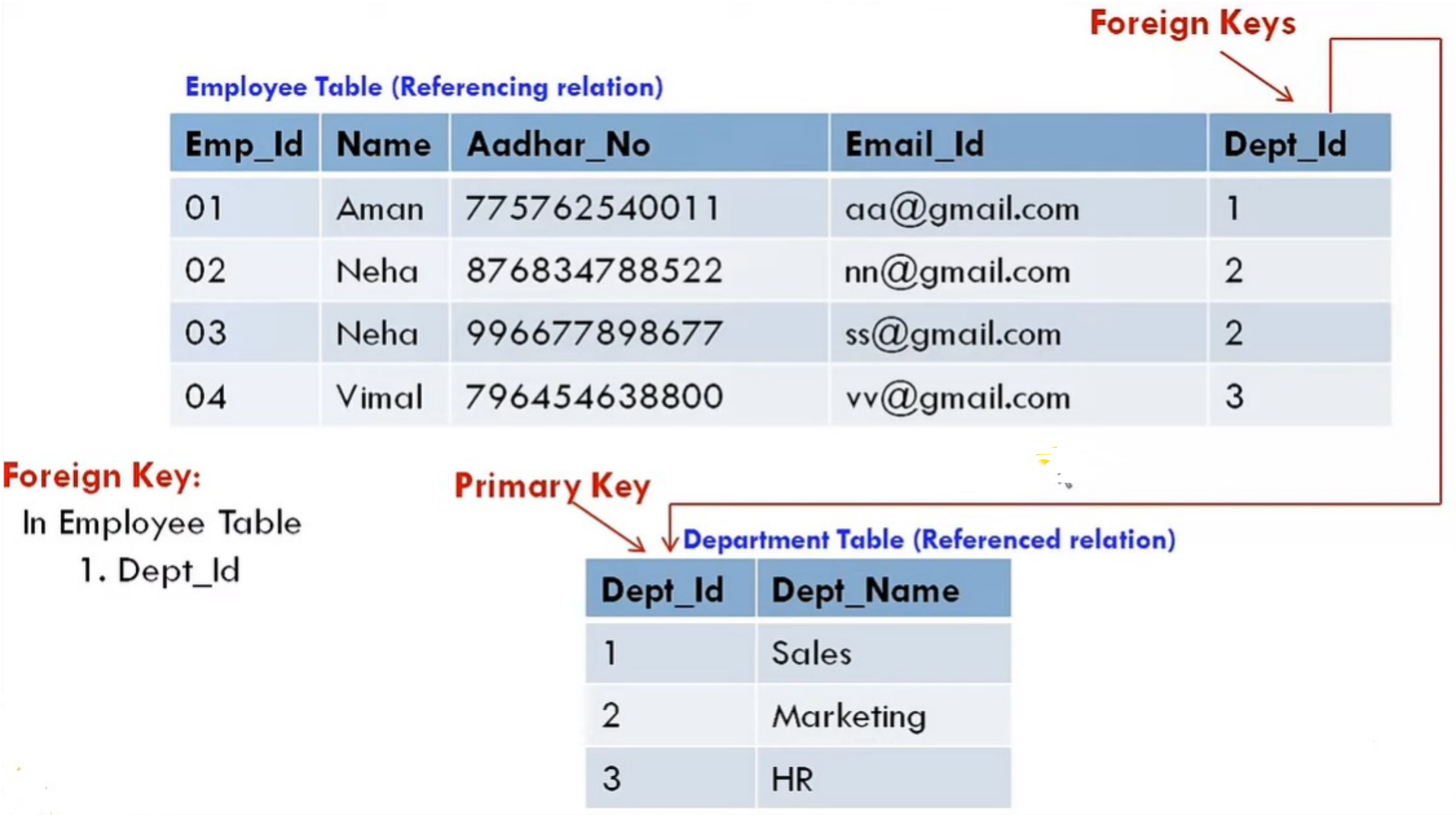
Candidate Key
(RollNo, Branch, Sem)

Alternate Key

Enroll_No	RollNo	Branch	Sem	SPI	Name	BL
190540107001	101	CE	3	8	Raju	0
180540107002	101	CE	5	7	Mitesh	1
190540106001	101	CI	3	6	Mayur	2
180540106002	101	CI	5	9	Nilesh	0

Foreign Key

- A foreign key is **used to link two relations** (tables).
- A foreign key is an **attribute** (or set of attributes) in one table that **refers to the primary key in another table**.
- A table containing the foreign key is called the child table/foreign table/referencing table, and the table containing the primary key is called the parent table/master table/primary table/referenced table.
- It is used to ensure referential integrity of the data.
- Foreign key can take only those values which are present in the primary key of the referenced table.
- Foreign key may have a name other than that of a primary key.
- Foreign key can take the null value.
- There is no restriction on a foreign key to be unique.
- One table may contain more than one foreign key.



Foreign Keys

Employee Table (Referencing relation)

Emp_Id	Name	Aadhar_No	Email_Id	Dept_Id
01	Aman	775762540011	aa@gmail.com	1
02	Neha	876834788522	nn@gmail.com	2
03	Neha	996677898677	ss@gmail.com	2
04	Vimal	796454638800	vv@gmail.com	3

Foreign Key:

In Employee Table

1. Dept_Id

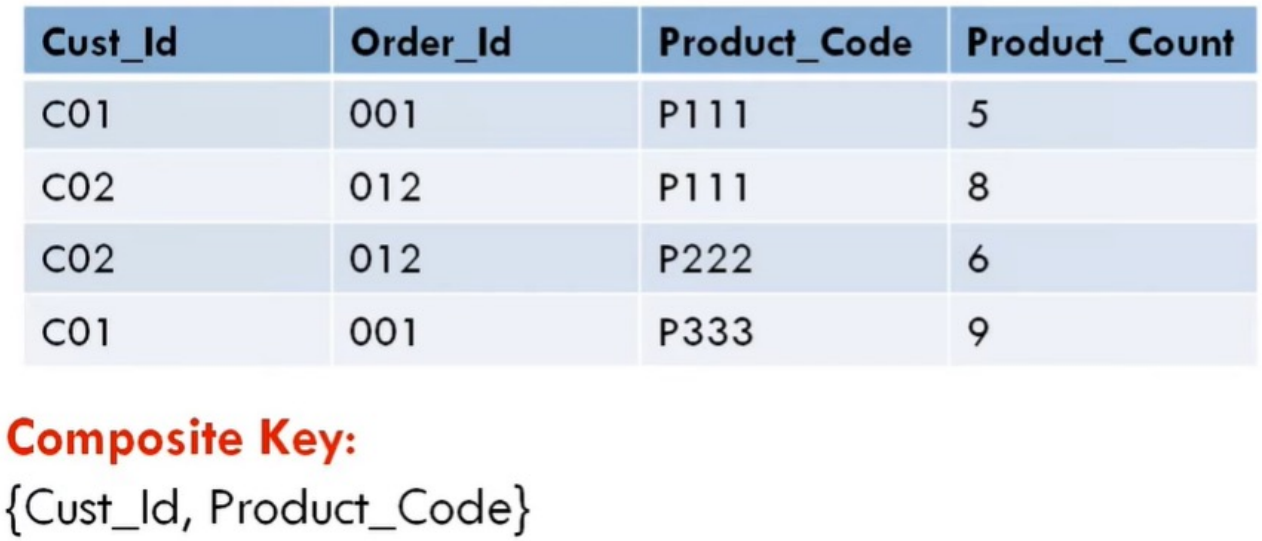
Primary Key

Department Table (Referenced relation)

Dept_Id	Dept_Name
1	Sales
2	Marketing
3	HR

Composite key

- A key that has more than one attribute is known as composite key.
- It is also known as compound key.



Cust_Id	Order_Id	Product_Code	Product_Count
C01	001	P111	5
C02	012	P111	8
C02	012	P222	6
C01	001	P333	9

Composite Key:

{Cust_Id, Product_Code}

Q. 1 Given table $R(A_1, A_2, A_3, \ldots\ldots..A_n)$ & given that candidate key of the relation = $\{A_1\}$, then how many number of super keys are possible?

C. K. = $\{A_1\}$

S. K. = $\{A_1\ A_2\}$

= $\{A_1\ A_3\}$

= $\{A_1\ A_2\ A_3\}$

= 2^{n-1}

	A1	A2	A3	A4	A5	..	..	..	..	An
#States	1	2	2	2	2	..	..	..	..	2
	$1 * 2^{n-1}$									

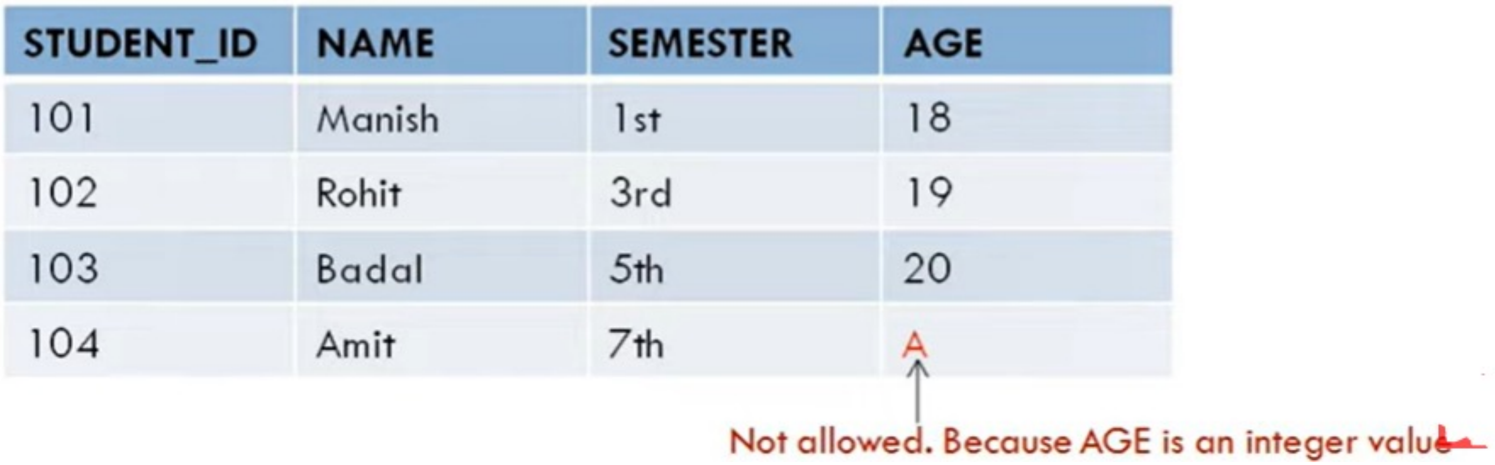
$R(A_1, A_2, A_3, \ldots\ldots..A_n)$		
1.	CK = $\{A_1, A_2\}$	2^{n-2}
2.	CK = $\{A_1, A_2, A_3\}$	2^{n-3}
3.	CK = $\{A_1\}\ \{A_2\}$	$2^{n-1} + 2^{n-1} - 2^{n-2}$
4.	CK = $\{A_1\}\ \{A_2, A_3\}$	$2^{n-1} + 2^{n-2} - 2^{n-3}$
5.	CK = $\{A_1, A_2\}\ \{A_{3,} A_4\}$	$2^{n-2} + 2^{n-2} - 2^{n-4}$
6.	CK = $\{A_1, A_2\}\ \{A_{1,} A_3\}$	$2^{n-2} + 2^{n-2} - 2^{n-3}$

Integrity Constraints

- Integrity constraints are a **set of rules** .
- They are used to ensure accuracy and consistency of the data in a relational database.
- Integrity constraints ensure that the data insertion, updation, and deletion made to the database by authorized users do not result in a loss of data consistency.
- Thus, integrity constraint is used to **guard against accidental damage** to the database.
- Constraints may apply to each attribute or relationships between tables.
- Various Integrity Constraints are:
  - Domain constraint (on attribute)
  - Key constraint (attribute)
  - Entity integrity constraint (attribute)
  - Referential integrity constraint (relationship)

Integrity Constraints: Domain constraint

- This constraint defines the domain or the valid set of values for an attribute. The data type of domain includes string, character, integer, time, date, currency, etc.
- Limits the data values of variables to a **specific set, range, or list of values**. (the value of the attribute must be available in the corresponding domain).
- The constraint can be applied for a single column or a group of columns.
- E.g. value of SPI should be between 0 to 10.



STUDENT_ID	NAME	SEMESTER	AGE
101	Manish	1st	18
102	Rohit	3rd	19
103	Badal	5th	20
104	Amit	7th	A

Not allowed. Because AGE is an integer value

Integrity Constraints: Key (Unique) constraint

- No two tuples in a relation have same value for all attributes. To ensure that we choose key constraint.

S_ID	S_NAME	YEAR	AGE
1	A	1	18
2	B	2	19
3	A	2	17
4	A	3	19
2	B	2	19

Not allowed

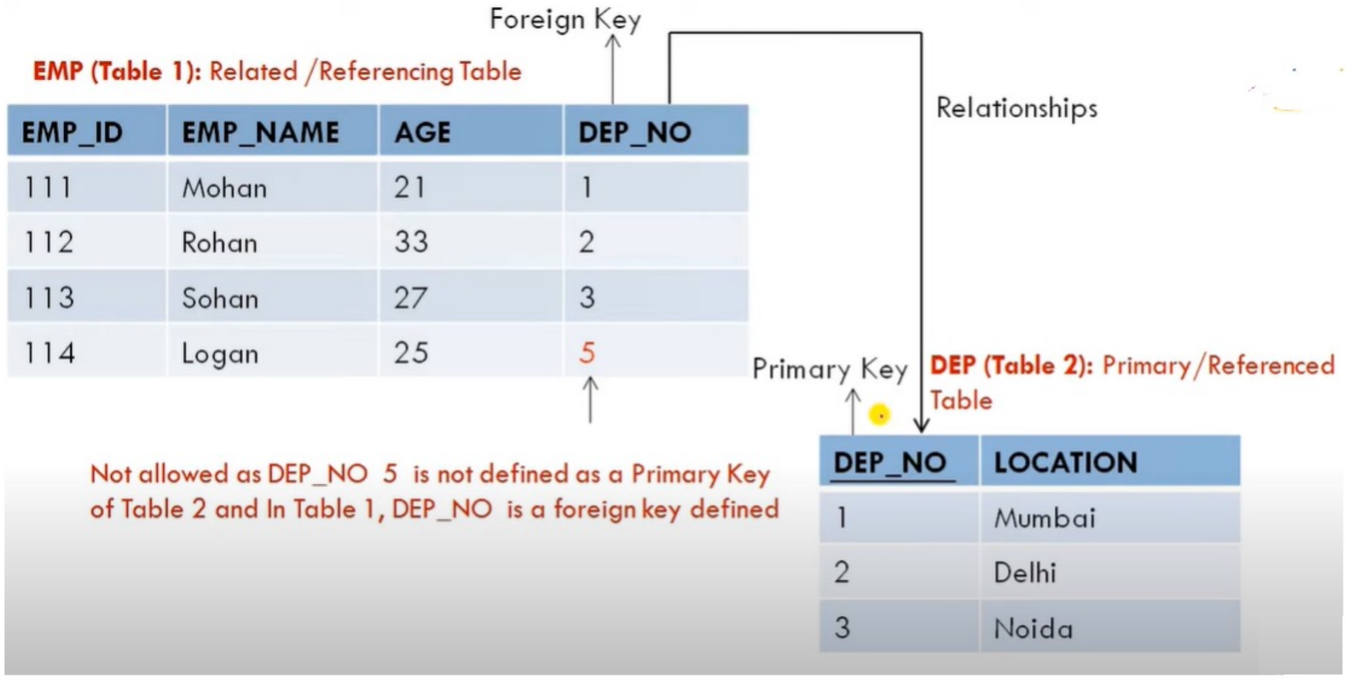
Integrity Constraints: Entity integrity constraint

- Primary key value can’t be NULL. Because it uniquely identifies individual rows in the table. If the primary key is null, then we cant identify those rows.

  E.g. enrollment_no column should have unique value as well as can’t be null.

Integrity Constraints: referential integrity constraint

- A referential integrity constraint is specified between two tables.
- referential integrity constraint is enforced when a foreign key references the primary key of a table.
- In the referential integrity constraints, if a foreign key column in table 1 refers to the primary key column of table 2, then every value of the foreign key column in table 1 must be null or be available in primary key column of table 2.



EMP (Table 1): Related /Referencing Table

EMP_ID	EMP_NAME	AGE	DEP_NO
111	Mohan	21	1
112	Rohan	33	2
113	Sohan	27	3
114	Logan	25	5

Not allowed as DEP_NO 5 is not defined as a Primary Key of Table 2 and In Table 1, DEP_NO is a foreign key defined

DEP (Table 2): Primary/Referenced Table

DEP_NO	LOCATION
1	Mumbai
2	Delhi
3	Noida

Integrity Constraints

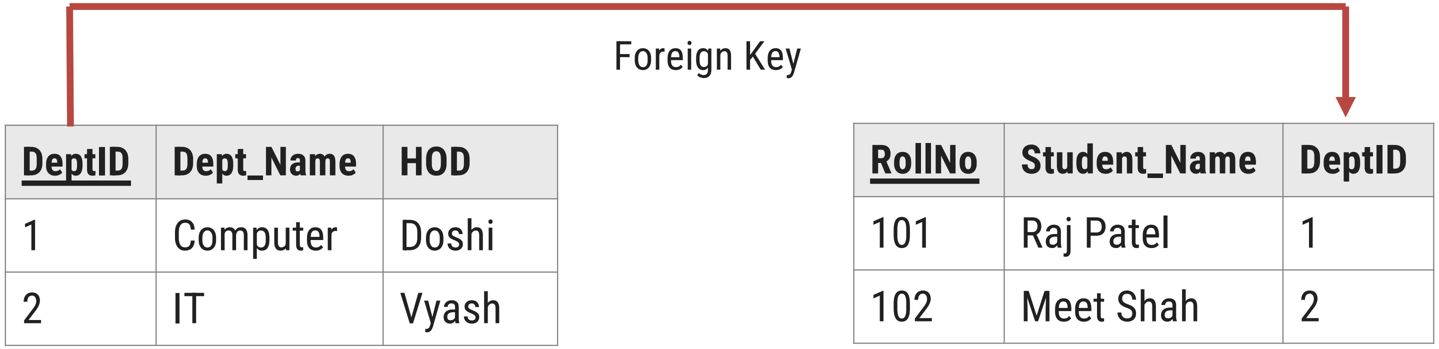
- Integrity constraints are a **set of rules**. It is used to **maintain the quality** of information.
- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- Thus, integrity constraint is used to **guard against accidental damage** to the database.
- Various Integrity Constraints are:
  - Check
  - Not null
  - Unique
  - Primary key
  - Foreign key

Integrity Constraints

- Check
  - This constraint defines a business rule on a column. All the rows in that column must satisfy this rule.
  - Limits the data values of variables to a **specific set, range, or list of values**.
  - The constraint can be applied for a single column or a group of columns.
  - E.g. value of SPI should be between 0 to 10.
- Not null
  - This constraint ensures all rows in the table contain a definite value for the column which is specified as not null. Which means a **null value** is not allowed.
  - E.g. name column should have some value.
- Unique
  - This constraint ensures that a column or a group of columns in each row have a **distinct (unique)** value.
  - A column(s) can have a null value but the values cannot be duplicated.
  - E.g. enrollmentno column should have unique value.

Integrity Constraints

- Primary key
  - This constraint defines a column or combination of columns which uniquely identifies each row in the table.
  - Primary key = **Unique key + Not null**
  - E.g. enrollment_no column should have unique value as well as can't be null.
- Foreign key (referential integrity constraint)
  - A referential integrity constraint (foreign key) is specified between two tables.
  - In the referential integrity constraints, if a foreign key column in table 1 refers to the primary key column of table 2, then every value of the foreign key column in table 1 must be null or be available in primary key column of table 2.



Foreign Key

DeptID	Dept_Name	HOD
1	Computer	Doshi
2	IT	Vyash

RollNo	Student_Name	DeptID
101	Raj Patel	1
102	Meet Shah	2