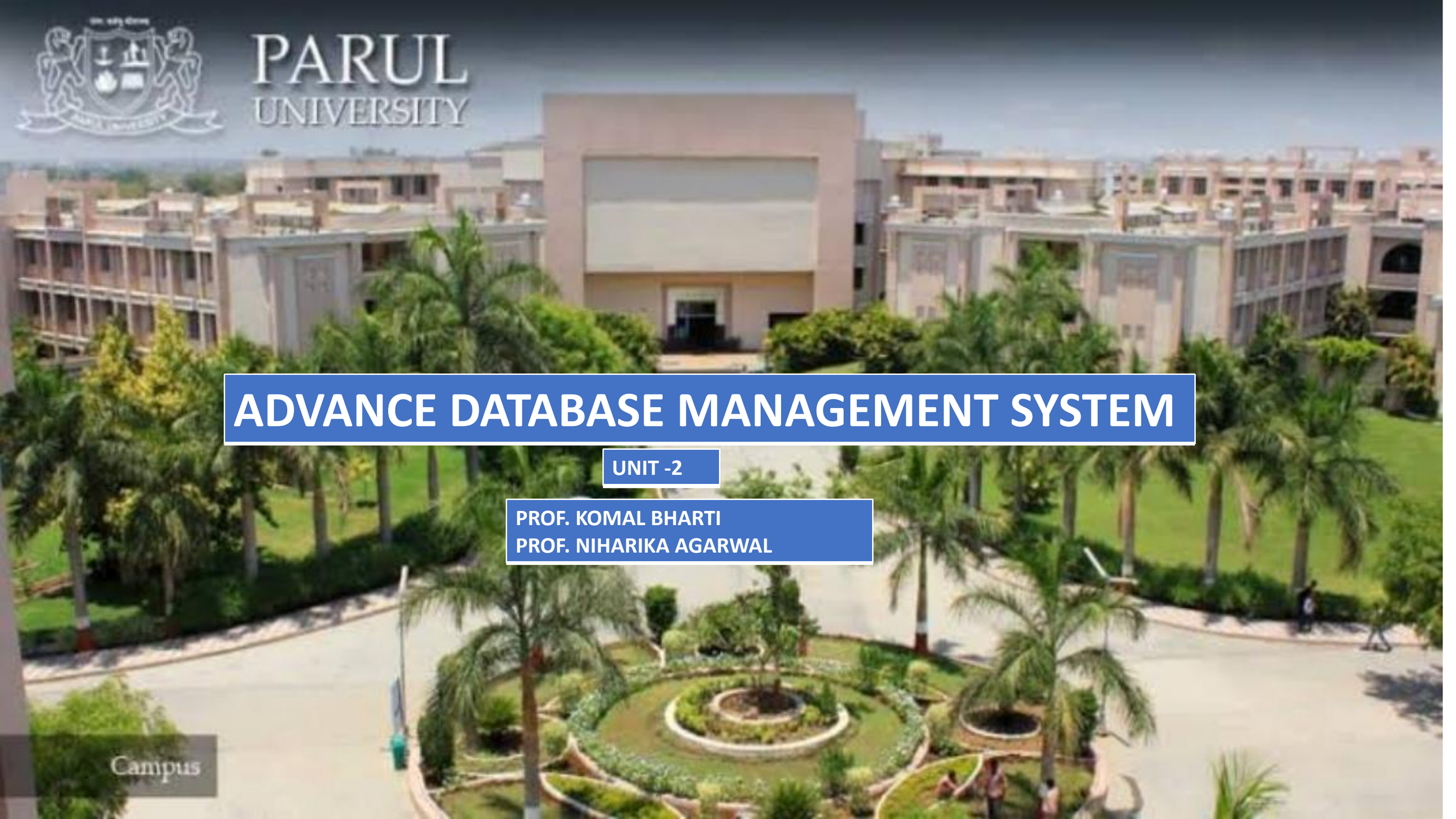


ADVANCE DATABASE MANAGEMENT SYSTEM

UNIT -2

PROF. KOMAL BHARTI
PROF. NIHARIKA AGARWAL

PARUL UNIVERSITY

What is **Relational Structure**?

The relational **Structure** uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name. Tables are also known as relations. The relational Structure is an example of a record-based model. Record-based models are so named because the database is structured in fixed-format records of several types. Each table contains records of a particular type. Each record type defines a fixed number of fields, or attributes. The columns of the table correspond to the attributes of the record type. The relational data model is the most widely used data model, and a vast majority of current database systems are based on the relational model.

PARUL UNIVERSITY

What is **Relational Structure**?

What is the Relational Structure?

The relational **Structure** represents how data is stored in Relational Databases. A relational database consists of a collection of tables, each of which is assigned a unique name. Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE, and AGE shown in the table.

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

important terminology of Relational Structure

1.Tables,
2.Rows,
3.Columns,
4.Entity sets,
5.Attributes,
6.Types of entities

important terminology of Relational Structure

TABLE:SQL Table is a collection of data which is organized in terms of rows and columns. In DBMS, the table is known as relation and row as a tuple.
Table is a simple form of data storage. A table is also considered as a convenient representation of relations.

EMP_ID	EMP_NAME	CITY	PHONE_NO
1	Kristen	Washington	7289201223
2	Anna	Franklin	9378282882
3	Jackson	Bristol	926478383 8
4	Kellan	California	7254728346
5	Ashley	Hawaii	9638482678

EXAMPLE OF TABLE CREATION

```
SQL> CREATE TABLE EMPLOYEE (
EMP_ID INT                    NOT NULL,
EMP_NAME VARCHAR (25) NOT NULL,
PHONE_NO INT                  NOT NULL,
ADDRESS CHAR (30),
PRIMARY KEY (ID)
);
```

important terminology of Relational Structure

Row: In the context of relational databases, a row represents a single record in a table. Each row contains values for each column defined in the table schema. In everyday database terminology, a row is the term most commonly used to refer to a horizontal entry in a table.

Working with Tuple in DBMS

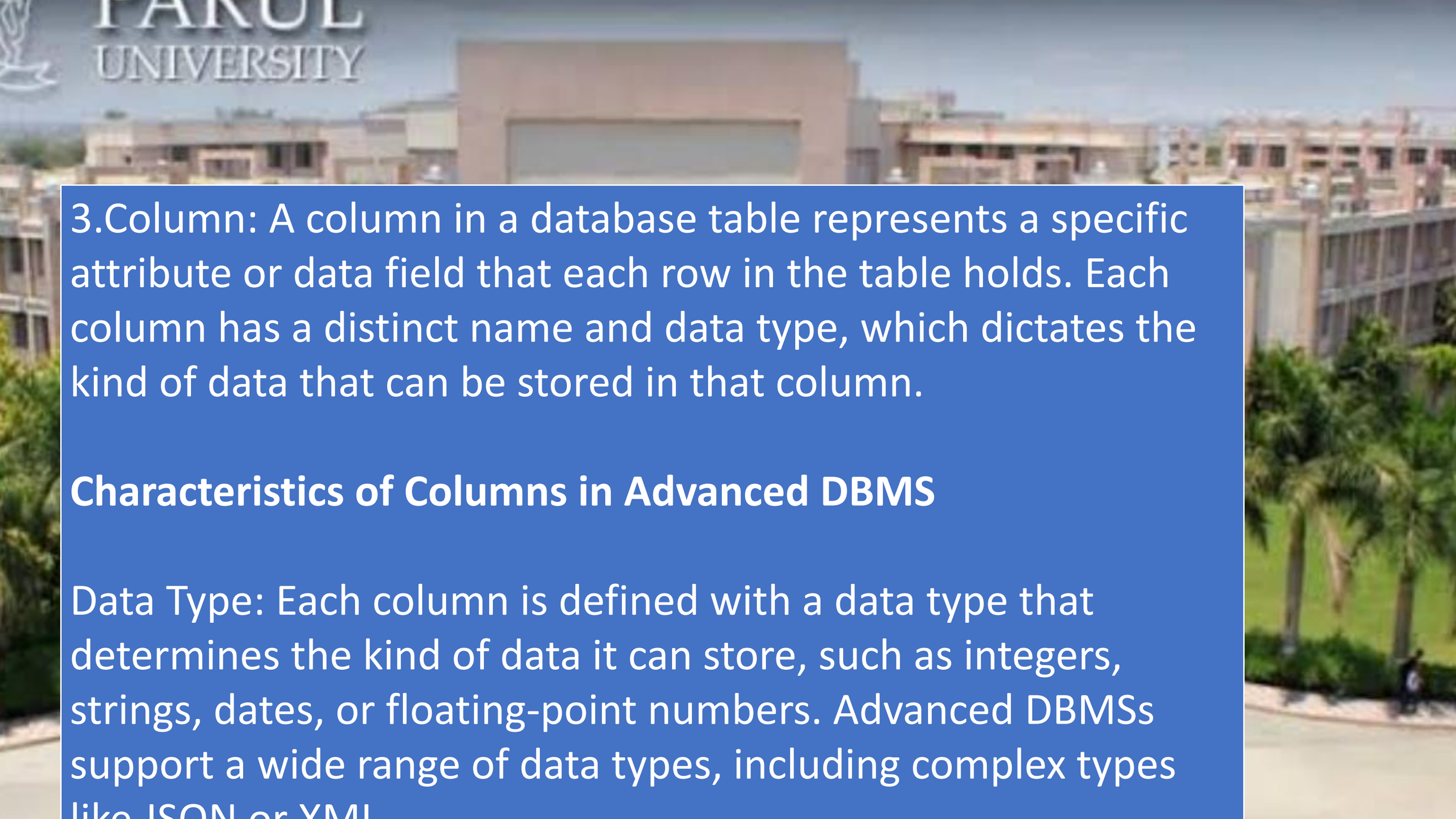
In a relational database, a relation is defined by a set of attributes and a set of tuples that have values for those attributes.

Example:

A relation called "CUSTOMER" might have attributes such as "customer_id", "first_name", "last_name", and "email". Each tuple in the relationship would have a unique value for the "customer_id" attribute and corresponding values for the other attributes, such as "John" for "first_name" and "Smith" for "last_name".

customer_id	first_name	last_name	email
1	John	Smith	abc@gmail.com
2	Abhishek	Bhosle	cde@gmail.com
3	Natasha	Witch	fgh@gmail.com

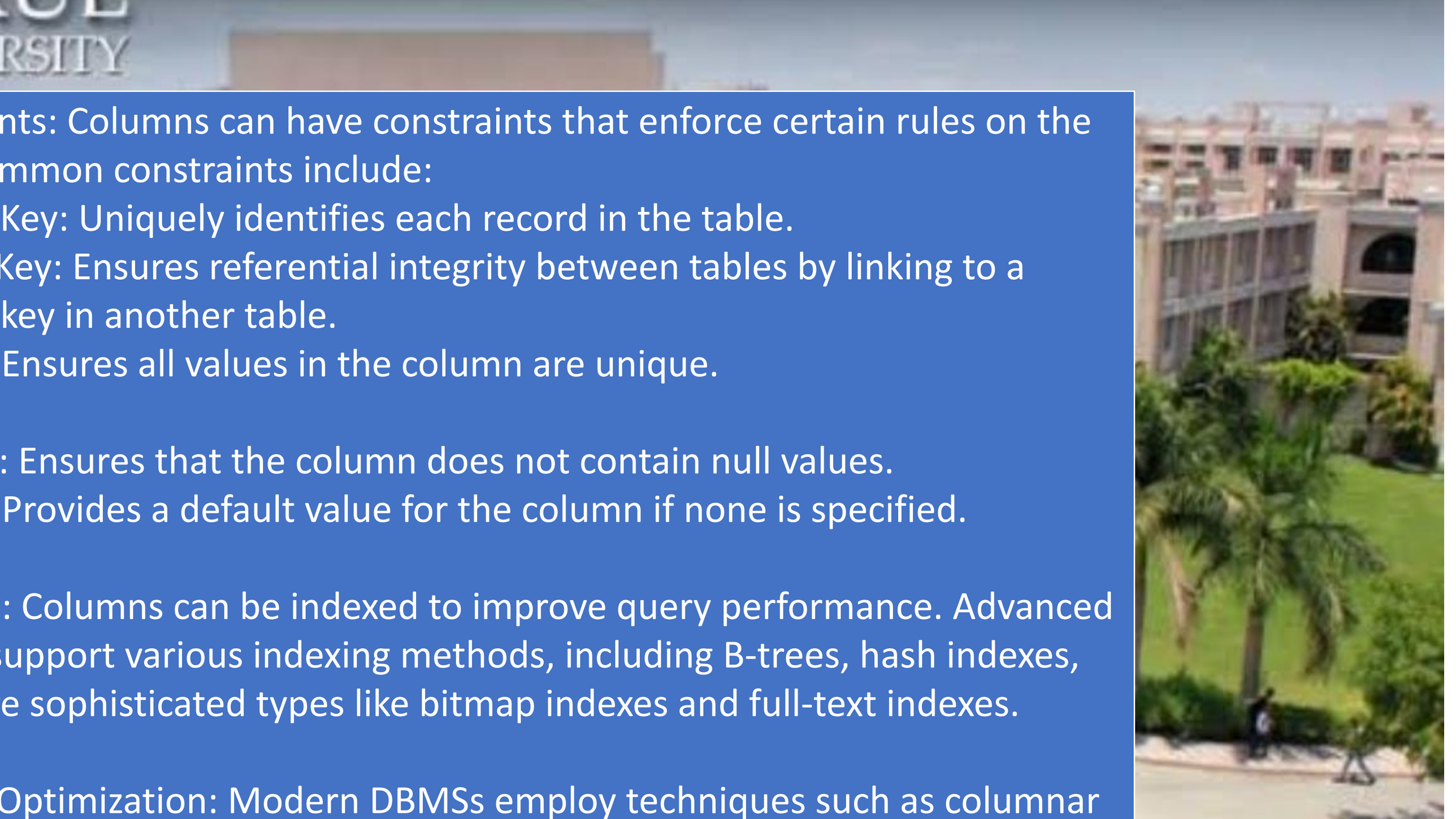
Tuples are also used in the process of normalization in a relational database. Normalization is the process of organizing data in a database to minimize data redundancy and improve data integrity. In the process of normalization, a relation is broken down into multiple smaller relations, each with a specific purpose and containing a specific set of attributes and tuples.



3.Column: A column in a database table represents a specific attribute or data field that each row in the table holds. Each column has a distinct name and data type, which dictates the kind of data that can be stored in that column.

Characteristics of Columns in Advanced DBMS

Data Type: Each column is defined with a data type that determines the kind of data it can store, such as integers, strings, dates, or floating-point numbers. Advanced DBMSs support a wide range of data types, including complex types like JSON or XML



nts: Columns can have constraints that enforce certain rules on the
mmon constraints include:
Key: Uniquely identifies each record in the table.
Key: Ensures referential integrity between tables by linking to a
key in another table.
Ensures all values in the column are unique.

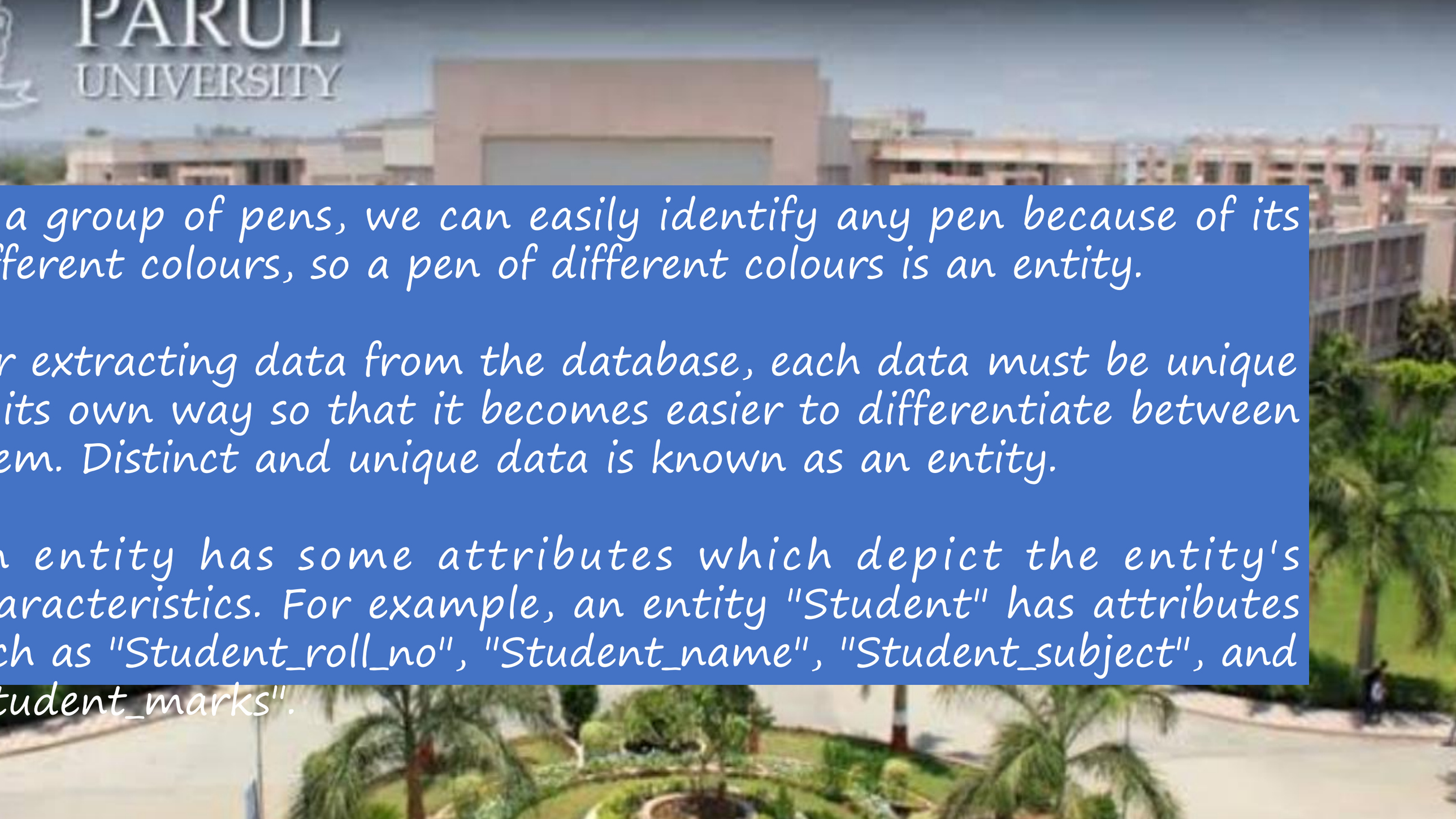
: Ensures that the column does not contain null values.
Provides a default value for the column if none is specified.

: Columns can be indexed to improve query performance. Advanced
support various indexing methods, including B-trees, hash indexes,
e sophisticated types like bitmap indexes and full-text indexes.

Optimization: Modern DBMSs employ techniques such as columnar

ntity:
n entity is referred to as an object or thing that exists in the
al world. For example, customer, car, pen, etc.
ntities are stored in the database, and they should be
istinguishable, i.e., they should be easily identifiable from the
roup. For example, a group of pens that are from the same
ompany cannot be identified, so they are only objects, but pens
ith different colours become unique and will be called an entity
ke a red pen, green pen, b

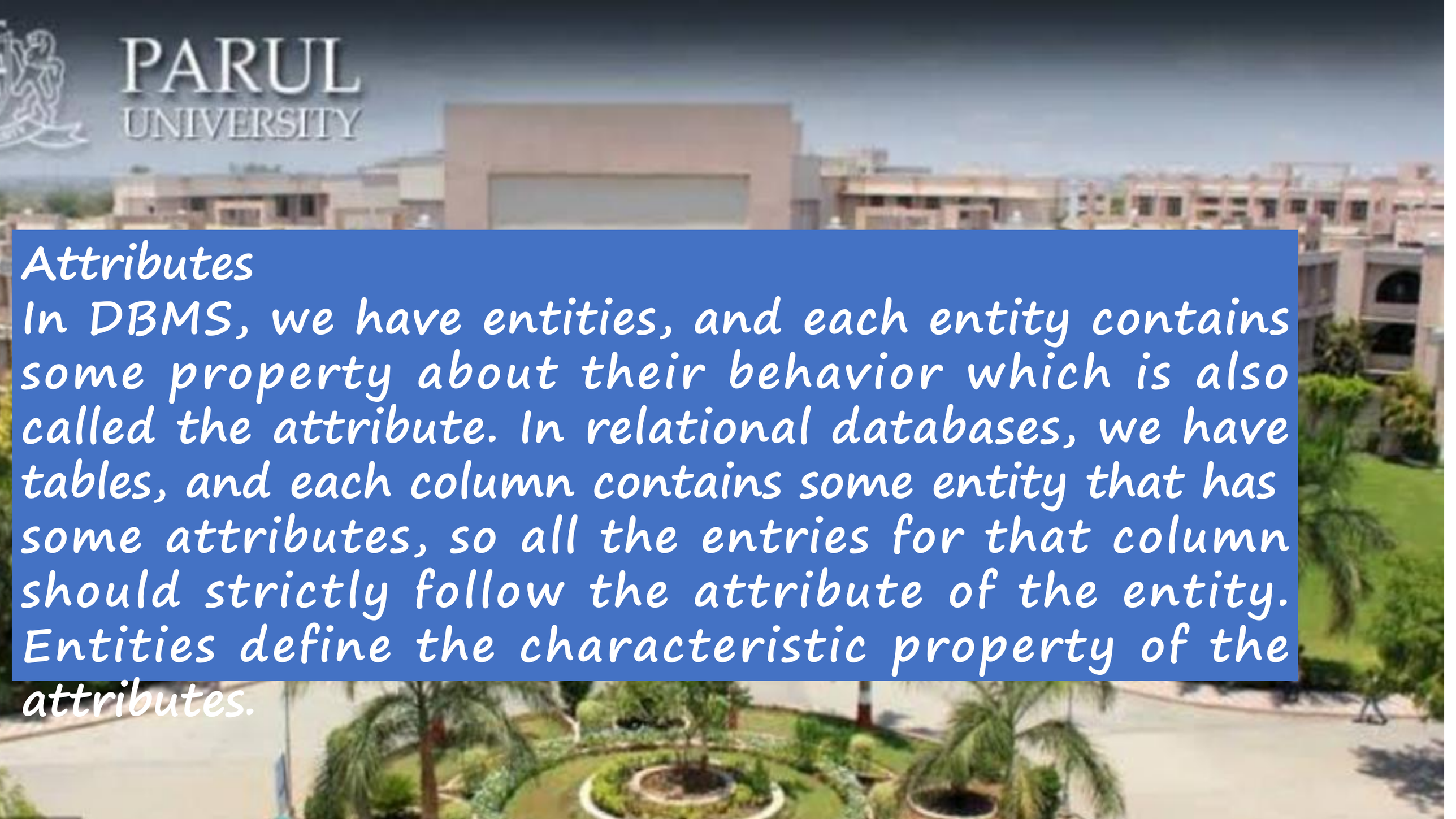
Database Management System (DBMS) is an essential tool to manage data, but do you know how important entities are in DBMS?
The role of the entity is the



a group of pens, we can easily identify any pen because of its fferent colours, so a pen of different colours is an entity.

r extracting data from the database, each data must be unique its own way so that it becomes easier to differentiate between em. Distinct and unique data is known as an entity.

n entity has some attributes which depict the entity's aracteristics. For example, an entity "Student" has attributes ch as "Student_roll_no", "Student_name", "Student_subject", and tudent_marks".



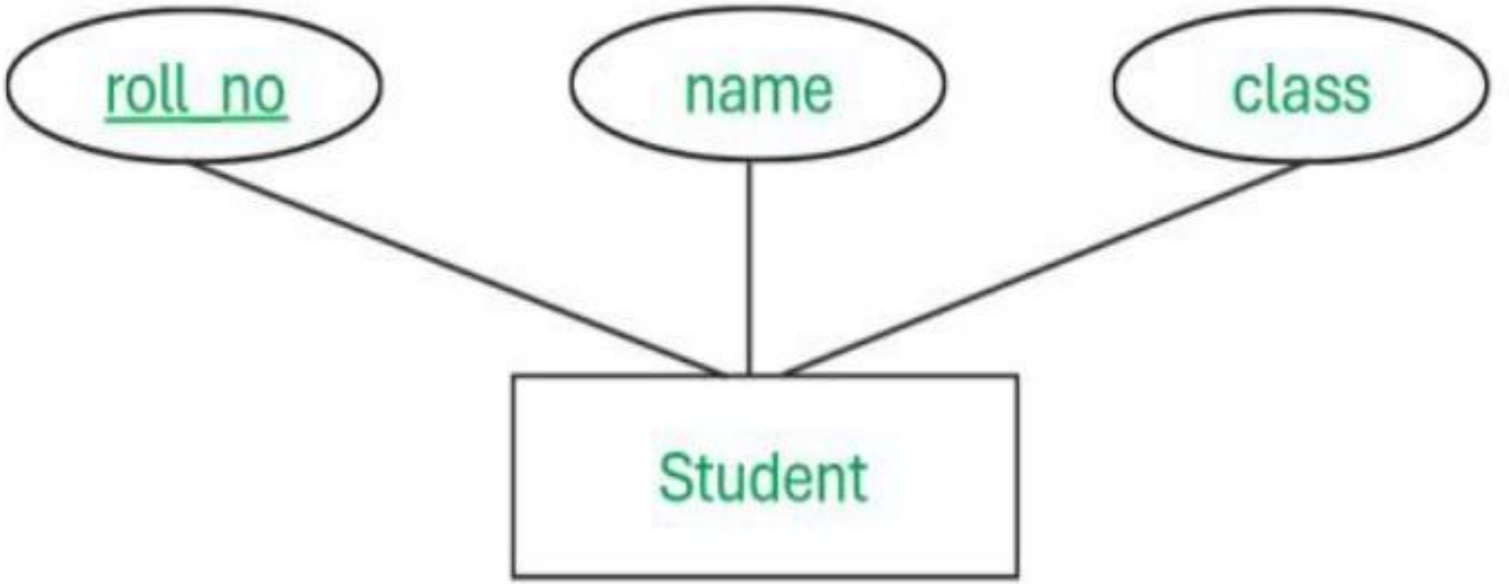
Attributes

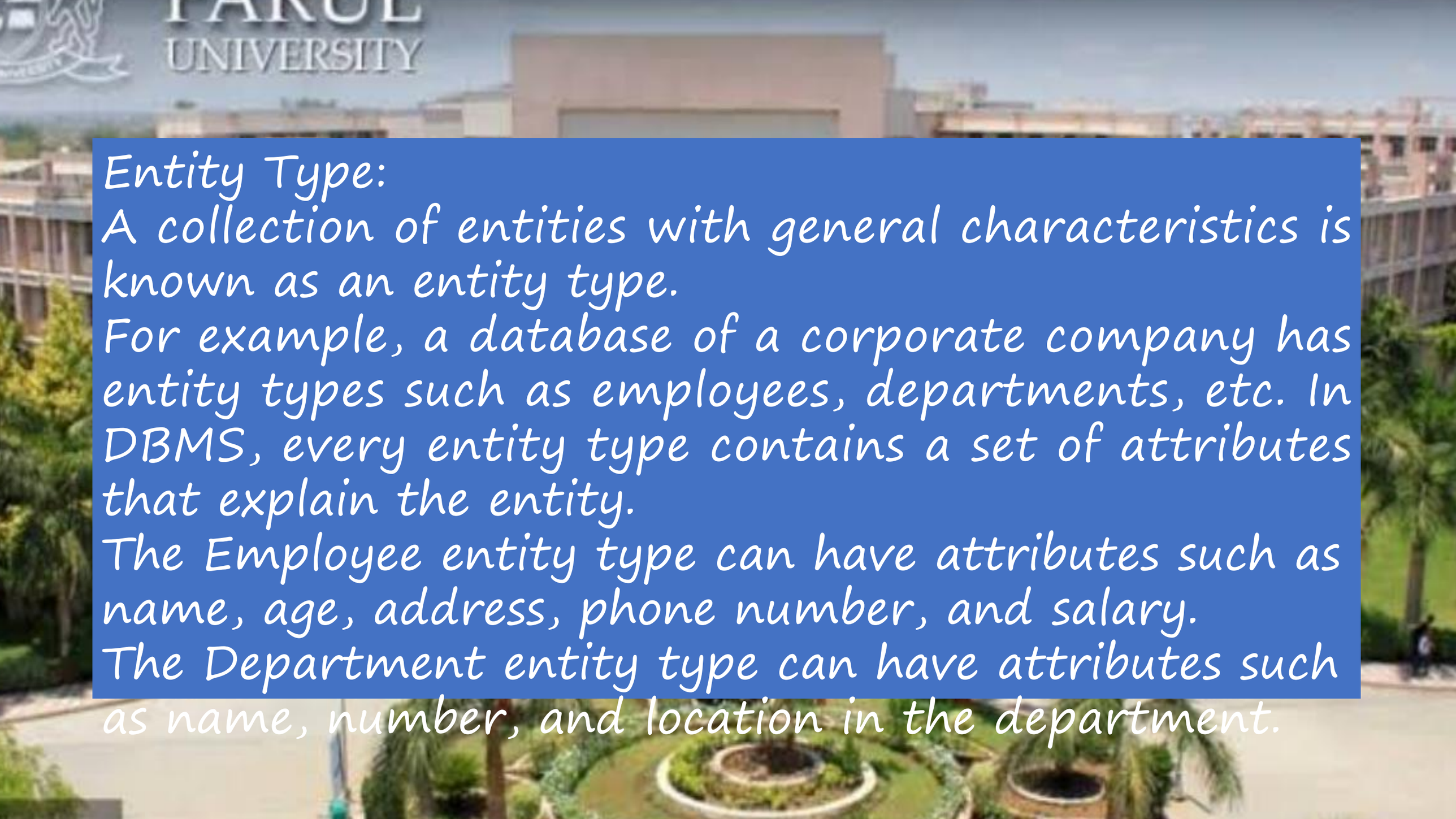
In DBMS, we have entities, and each entity contains some property about their behavior which is also called the attribute. In relational databases, we have tables, and each column contains some entity that has some attributes, so all the entries for that column should strictly follow the attribute of the entity. Entities define the characteristic property of the attributes.

ributes are properties or characteristics of an entity. ributes are used to describe the entity. The attribute is ning but a piece of data that gives more information about entity. Attributes are used to distinguish one entity from other entity. Attributes help to categorize the entity and entity can be easily retrieved and manipulate the entity. ributes can help the database to be more structural and archical. An entity with no attribute is of no use in the abase.

mple

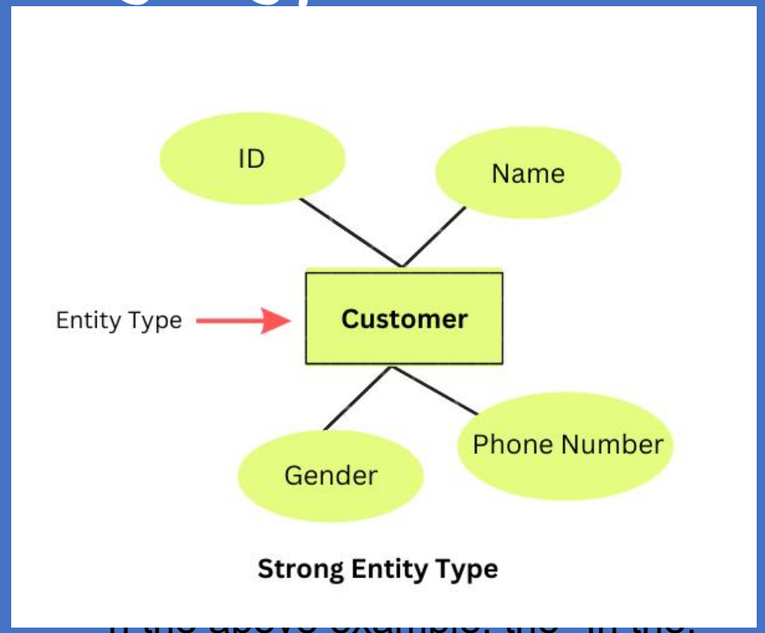
's take the student as an entity. Students will have tiple attributes such as roll number, name, and class. se attributes are used to describe the student in more il





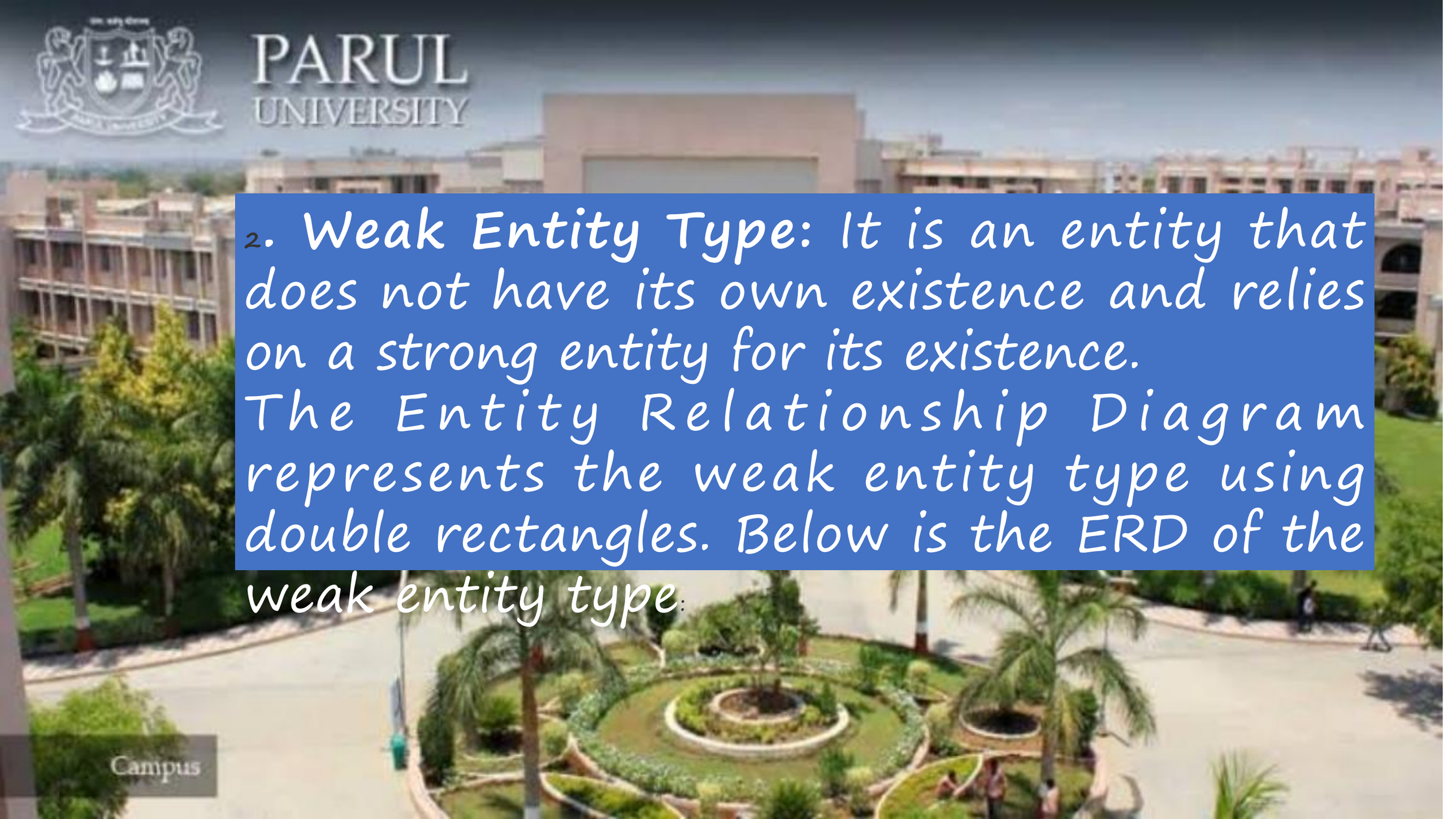
Entity Type:
A collection of entities with general characteristics is known as an entity type.
For example, a database of a corporate company has entity types such as employees, departments, etc. In DBMS, every entity type contains a set of attributes that explain the entity.
The Employee entity type can have attributes such as name, age, address, phone number, and salary.
The Department entity type can have attributes such as name, number, and location in the department.

1. **Strong Entity Type:** It is an entity that has its own existence and is independent. The entity relationship diagram represents a strong entity type with the help of a single rectangle. Below is the ERD of the strong entity type:



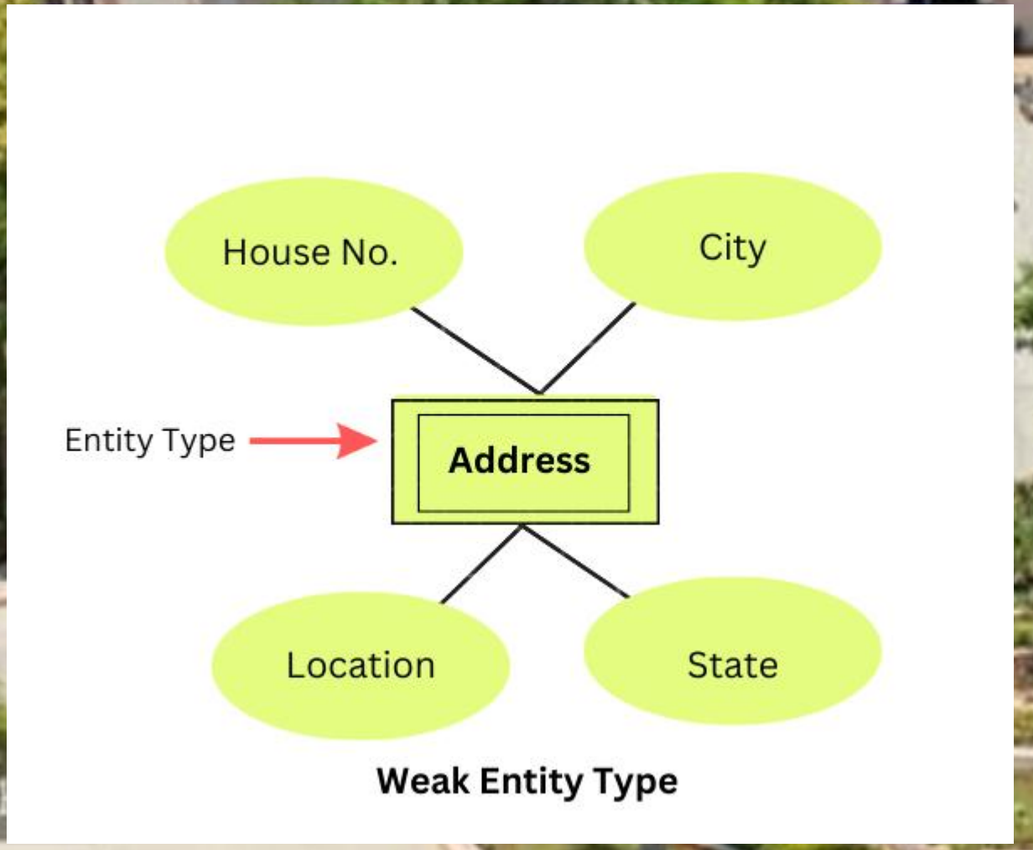
Strong Entity Type

In the above example, the "ID" in the

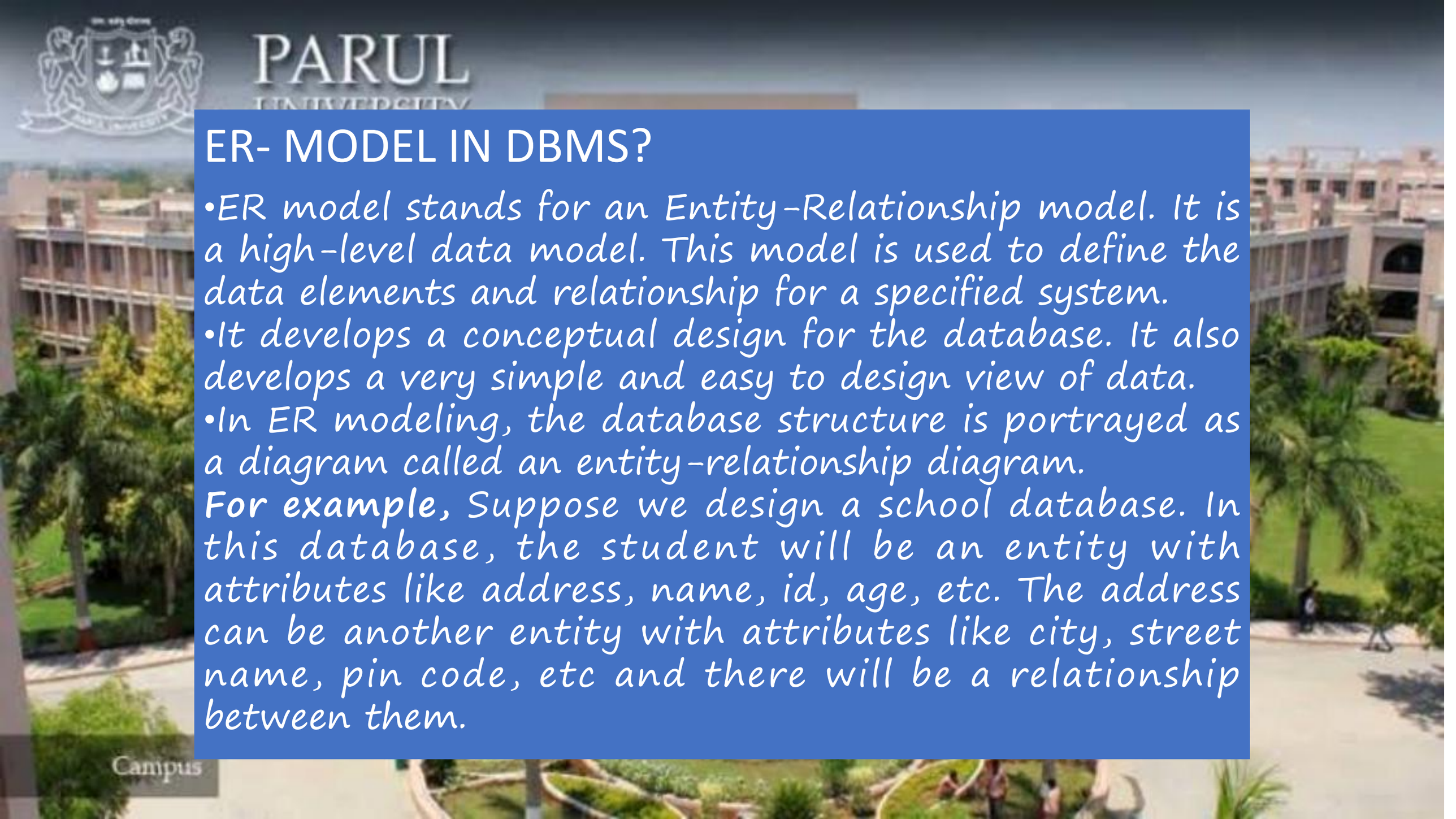


2. **Weak Entity Type:** It is an entity that does not have its own existence and relies on a strong entity for its existence.
The Entity Relationship Diagram represents the weak entity type using double rectangles. Below is the ERD of the weak entity type:

In the above example, the "Customer" is the entity type with attributes such as ID, Name, Gender, and Phone Number.
Customer is a strong entity type as it has a unique ID for each customer.



In the above example, "Address" is a weak entity type with attributes such as House No., City, Location, and State.
The relationship between a strong and a weak entity type is known as an identifying relationship.
Using a double diamond, the Entity-Relationship Diagram represents a relationship between the strong and the weak entity type.



ER- MODEL IN DBMS?

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

For example, Suppose we design a school database. In this database, the student will be an entity with attributes like address, name, id, age, etc. The address can be another entity with attributes like city, street name, pin code, etc and there will be a relationship between them.

PARUL UNIVERSITY

ER- MODEL IN DBMS?

Why Use ER Diagrams In DBMS?

ER diagrams represent the E-R model in a database, making them easy to convert into relations (tables).

ER diagrams provide the purpose of real-world modeling of objects which makes them intently useful.

ER diagrams require no technical knowledge and no hardware support.

These diagrams are very easy to understand and easy to create even for a naive user.

It gives a standard solution for visualizing the data logically.

Campus

ER- MODEL IN DBMS?

Symbols Used in ER Model

ER Model is used to model the logical view of the system from a data perspective which consists of these symbols:

Rectangles: Rectangles represent Entities in the ER Model.

Ellipses: Ellipses represent Attributes in the ER Model.

Diamond: Diamonds represent Relationships among Entities.

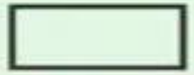
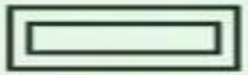
Lines: Lines represent attributes to entities and entity sets with other relationship types.

Double Ellipse: Double Ellipses represent Multi-Valued Attributes.

Double Rectangle: Double Rectangle represents a Weak Entity.

PARUL UNIVERSITY

ER- MODEL IN DBMS?

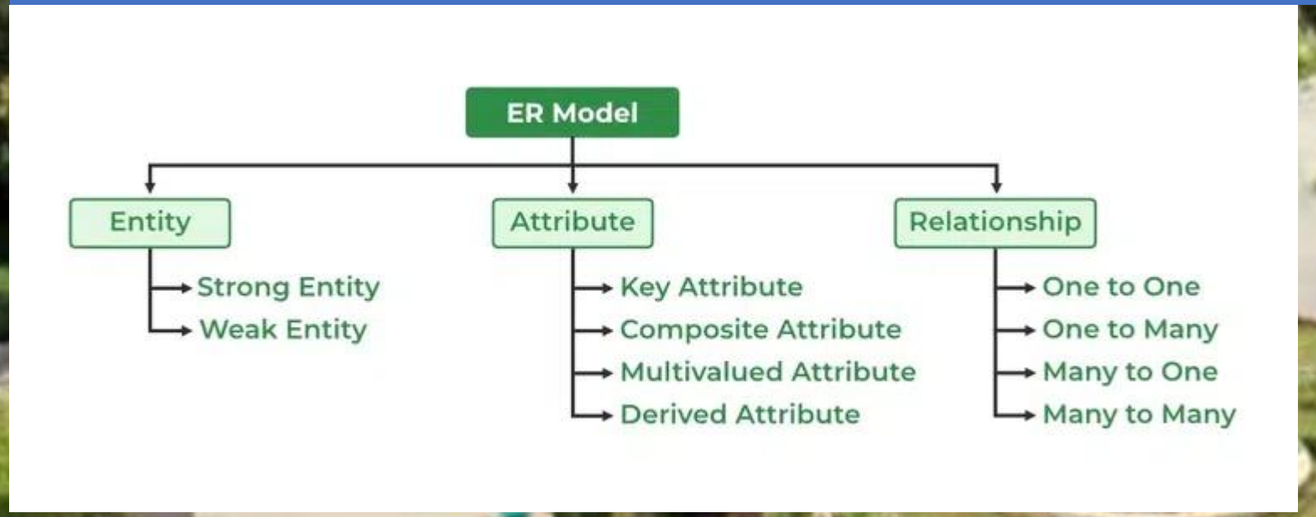
Figures	Symbols	Represents
Rectangle		Entities in ER Model
Ellipse		Attributes in ER Model
Diamond		Relationships among Entities
Line		Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse		Multi-Valued Attributes
Double Rectangle		Weak Entity

PARUL UNIVERSITY

ER- MODEL IN DBMS?

Components of ER Diagram

ER Model consists of Entities, Attributes, and Relationships among Entities in a Database System.



Campus

PARUL UNIVERSITY

ER- MODEL IN DBMS?

What is Entity?

An Entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.

What is Entity Set?

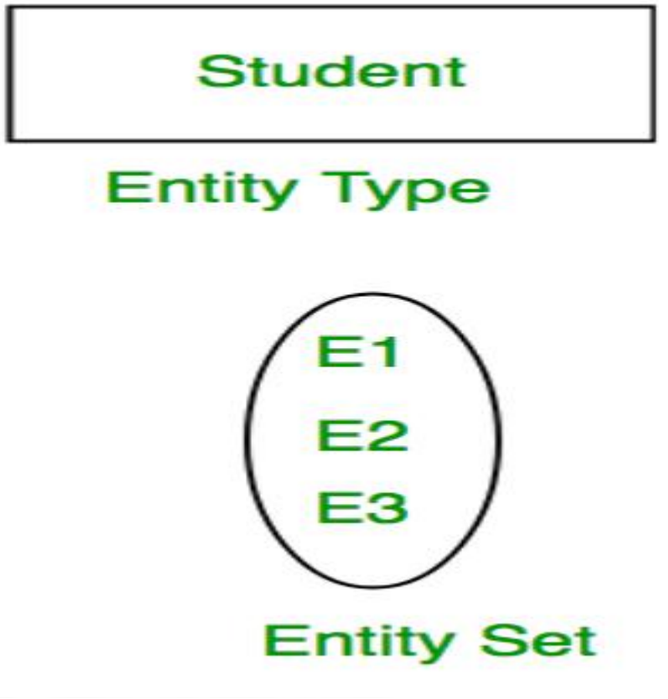
An Entity is an object of Entity Type and a set of all entities is called an entity set. For Example, E1 is an entity having Entity Type Student and the set of all students is called Entity Set. In ER diagram, Entity Type is represented as:

Campus

PARUL UNIVERSITY

ER- MODEL IN DBMS?

We can represent the entity set in ER Diagram but can't represent entity in ER Diagram because entity is row and column in the relation and ER Diagram is graphical representation of data.



Campus

PARUL UNIVERSITY

ER- MODEL IN DBMS?

Types of Entity

There are two types of entity:

1. Strong Entity

A Strong Entity is a type of entity that has a key Attribute. Strong Entity does not depend on other Entity in the Schema. It has a primary key, that helps in identifying it uniquely, and it is represented by a rectangle. These are called Strong Entity Types.

Campus

PARUL UNIVERSITY

ER- MODEL IN DBMS?

2. Weak Entity

An Entity type has a key attribute that uniquely identifies each entity in the entity set. But some entity type exists for which key attributes can't be defined. These are called Weak Entity types

For Example, A company may store the information of dependents (Parents, Children, Spouse) of an Employee. But the dependents can't exist without the employee. So Dependent will be a **Weak Entity Type** and Employee will be Identifying Entity type for Dependent, which means it is **Strong Entity Type**.

A weak entity type is represented by a Double Rectangle. The participation of weak entity types is always total. The relationship between the weak entity type and its identifying strong entity type is called identifying relationship and it is represented by a double diamond.

PARUL UNIVERSITY

What is Attributes?

Attributes are the properties that define the entity type. For example, Roll_No, Name, DOB, Age, Address, and Mobile_No are the attributes that define entity type Student. In ER diagram, the attribute is represented by an oval.

Campus

PARUL UNIVERSITY

Types of Attributes

1. Key Attribute

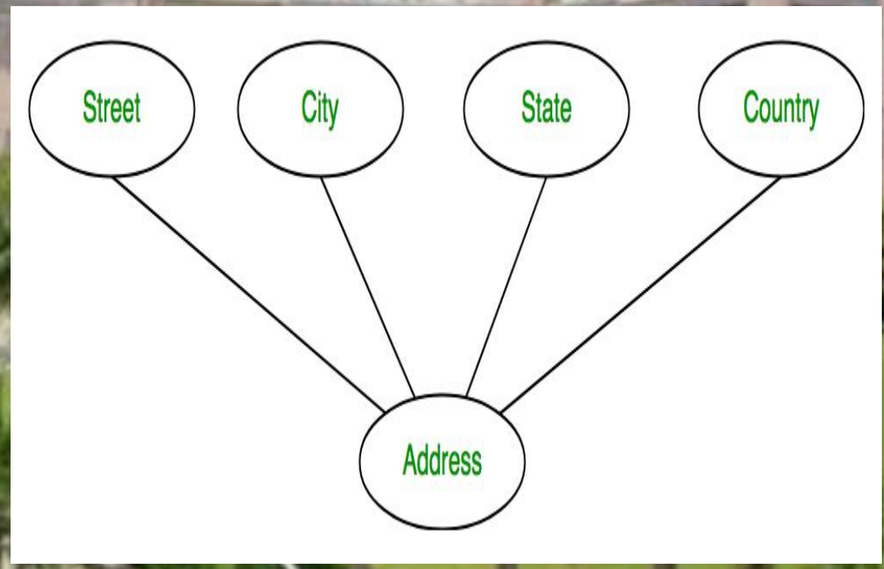
The attribute which **uniquely identifies each entity** in the entity set is called the key attribute. For example, Roll_No will be unique for each student. In ER diagram, the key attribute is represented by an oval with underlying lines.

Campus

PARUL UNIVERSITY

2. Composite Attribute

An attribute **composed of many other attributes** is called a composite attribute. For example, the Address attribute of the student Entity type consists of Street, City, State, and Country. In ER diagram, the composite attribute is represented by an oval comprising of ovals.



Campus

PARUL UNIVERSITY

3. Multivalued Attribute

An attribute consisting of more than one value for a given entity. For example, Phone_No,address (can be more than one for a given student). In ER diagram, a multivalued attribute is represented by a double oval.

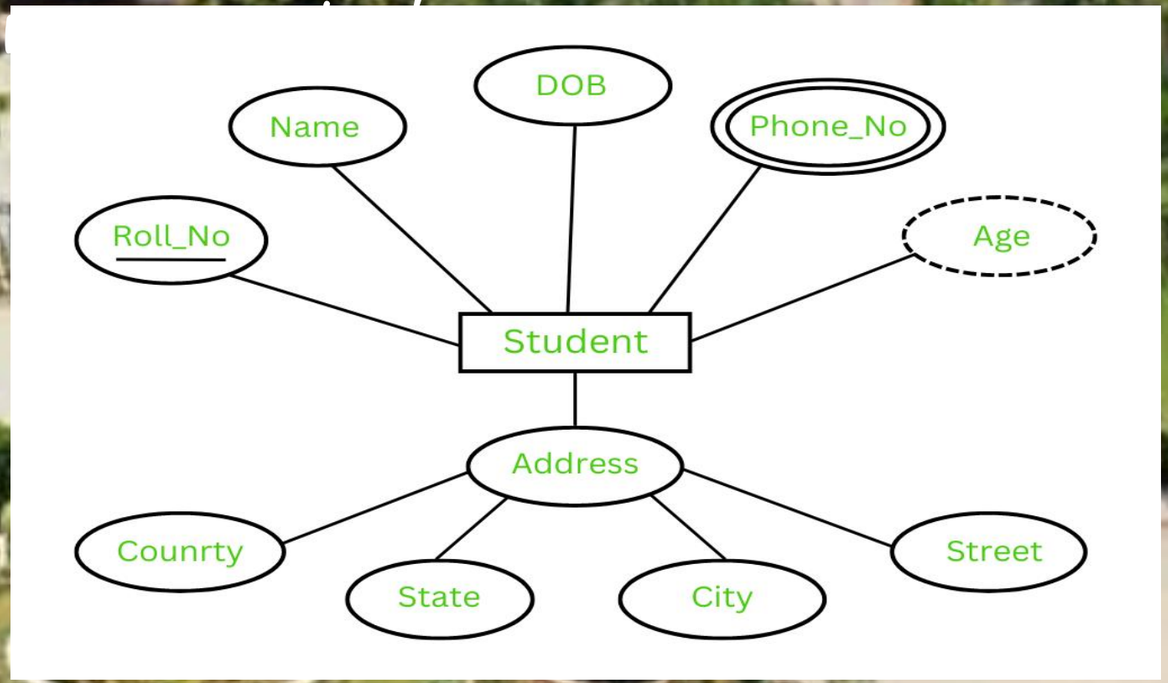
Campus

PARUL UNIVERSITY

4. Derived Attribute

An attribute that can be derived from other attributes of the entity type is known as a derived attribute. e.g.; Age (can be derived from DOB). In ER diagram, the derived attribute is repre… a dashed oval.

The Complete Entity Type Student with its Attributes can be …



PARUL UNIVERSITY

Relationship Type and Relationship Set

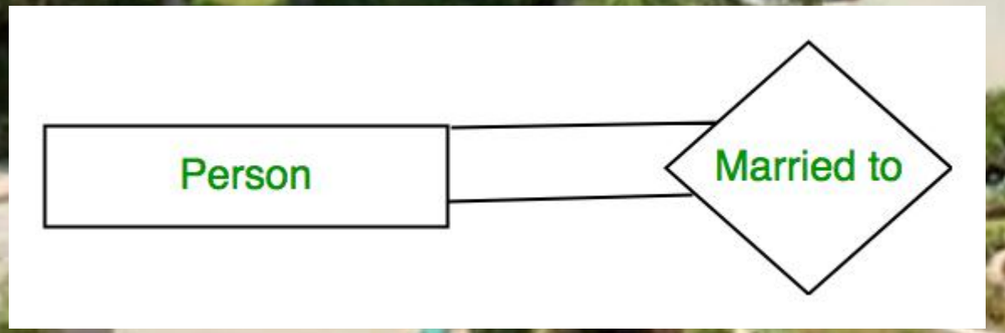
A Relationship Type represents the association between entity types. For example, 'Enrolled in' is a relationship type that exists between entity type Student and Course. In ER diagram, the relationship type is represented by a diamond and connecting the entities with lines.

Campus

Degree of a Relationship Set

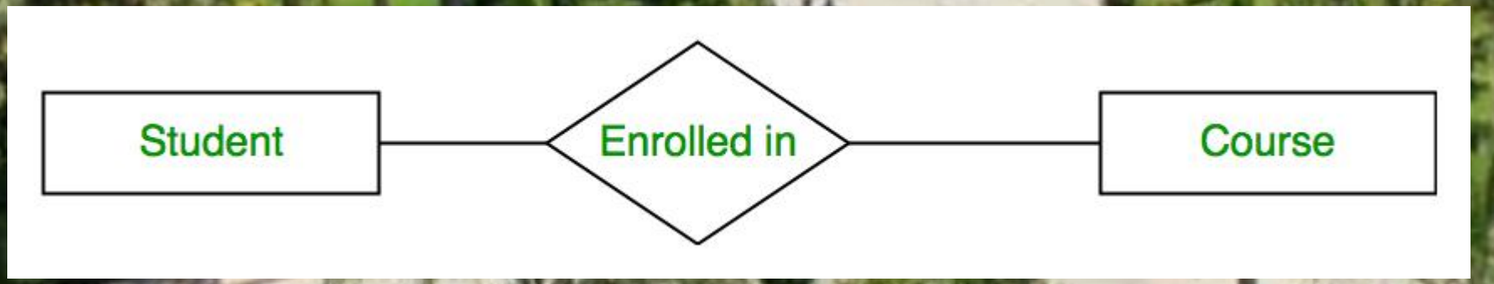
The number of different entity sets participating in a relationship set is called the degree of a relationship set.

1. Unary Relationship: When there is only ONE entity set participating in a relation, the relationship is called a unary relationship. For example, one person is married to only one person.



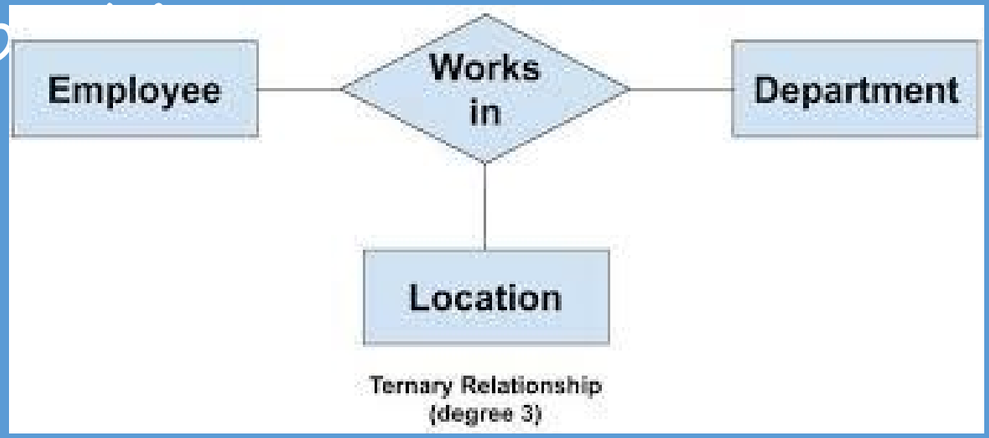
PARUL UNIVERSITY

2. Binary Relationship: When there are TWO entities set participating in a relationship, the relationship is called a binary relationship. For example, a Student is enrolled in a Course.

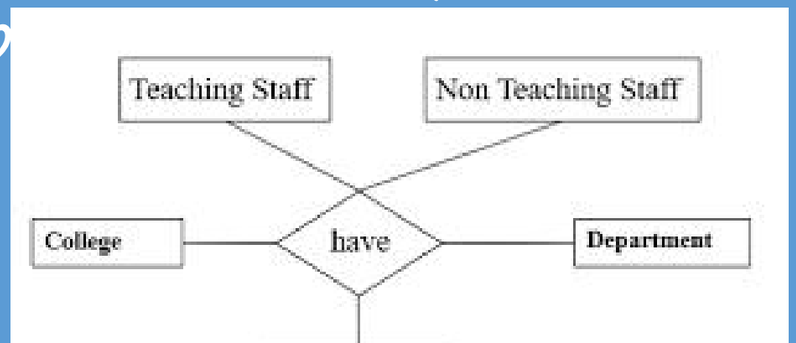


Campus

3. Ternary Relationship: When there are three entity sets participating in a relationship, the relationship is called a ternary relatio



4. N-ary Relationship: When there are n entities set participating in a relationship, the relationship is called an n-ary relationship

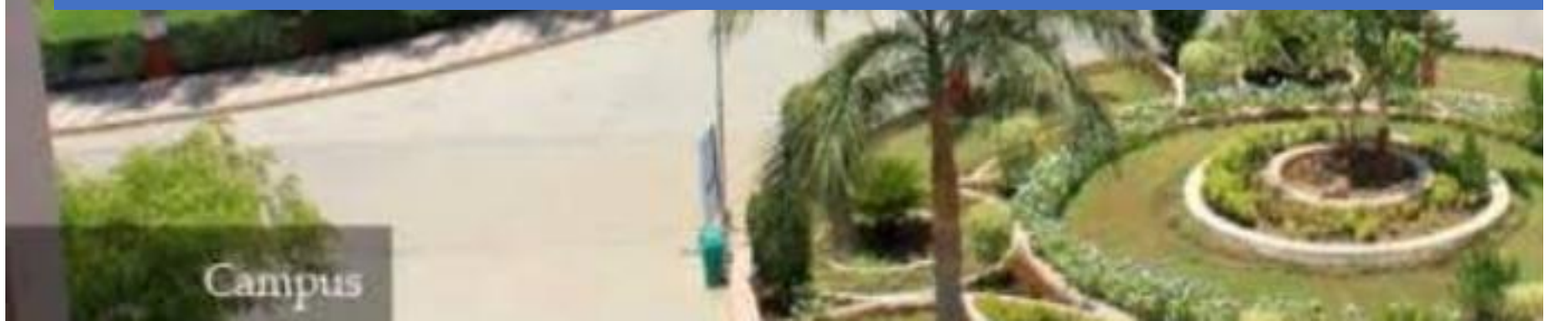


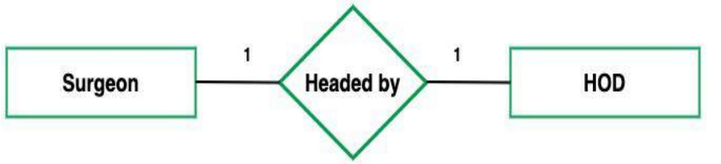
What is Cardinality?

The number of times an entity of an entity set participates in a relationship set is known as cardinality Cardinality can be of different types:

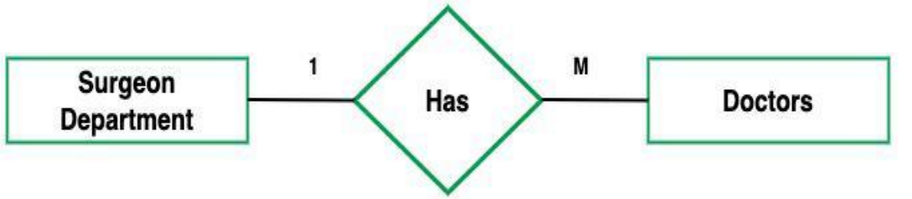
1. One-to-One: When each entity in each entity set can take part only once in the relationship, the cardinality is one-to-one. Let us assume that a male can marry one female and a female can marry one male. So the relationship will be one-to-one.

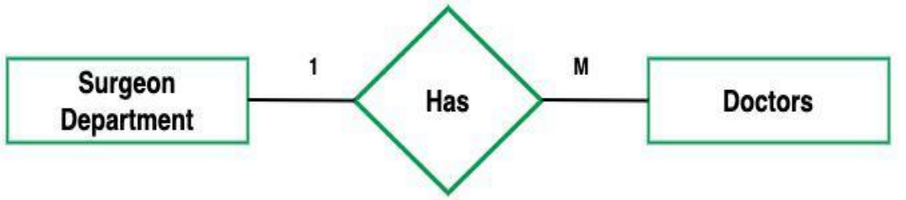
the total number of tables that can be u

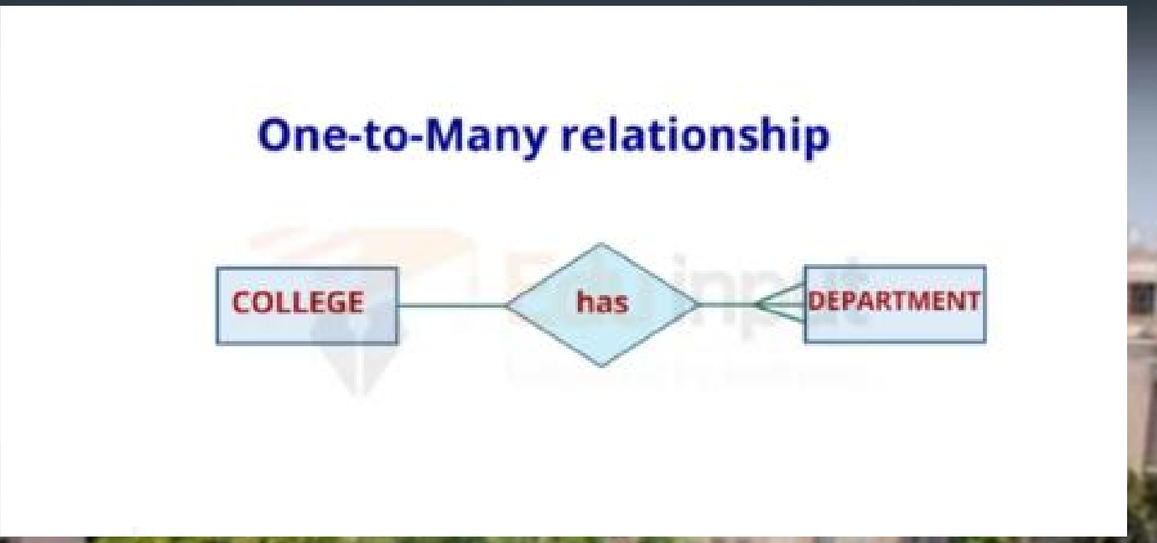




2. One-to-Many: In one-to-many mapping as well where each entity can be related to more than one entity and the total number of tables that can be used in this is 2. Let us assume that one surgeon department can accommodate many doctors. So the Cardinality will be 1 to M. It means one department has many Doctors.

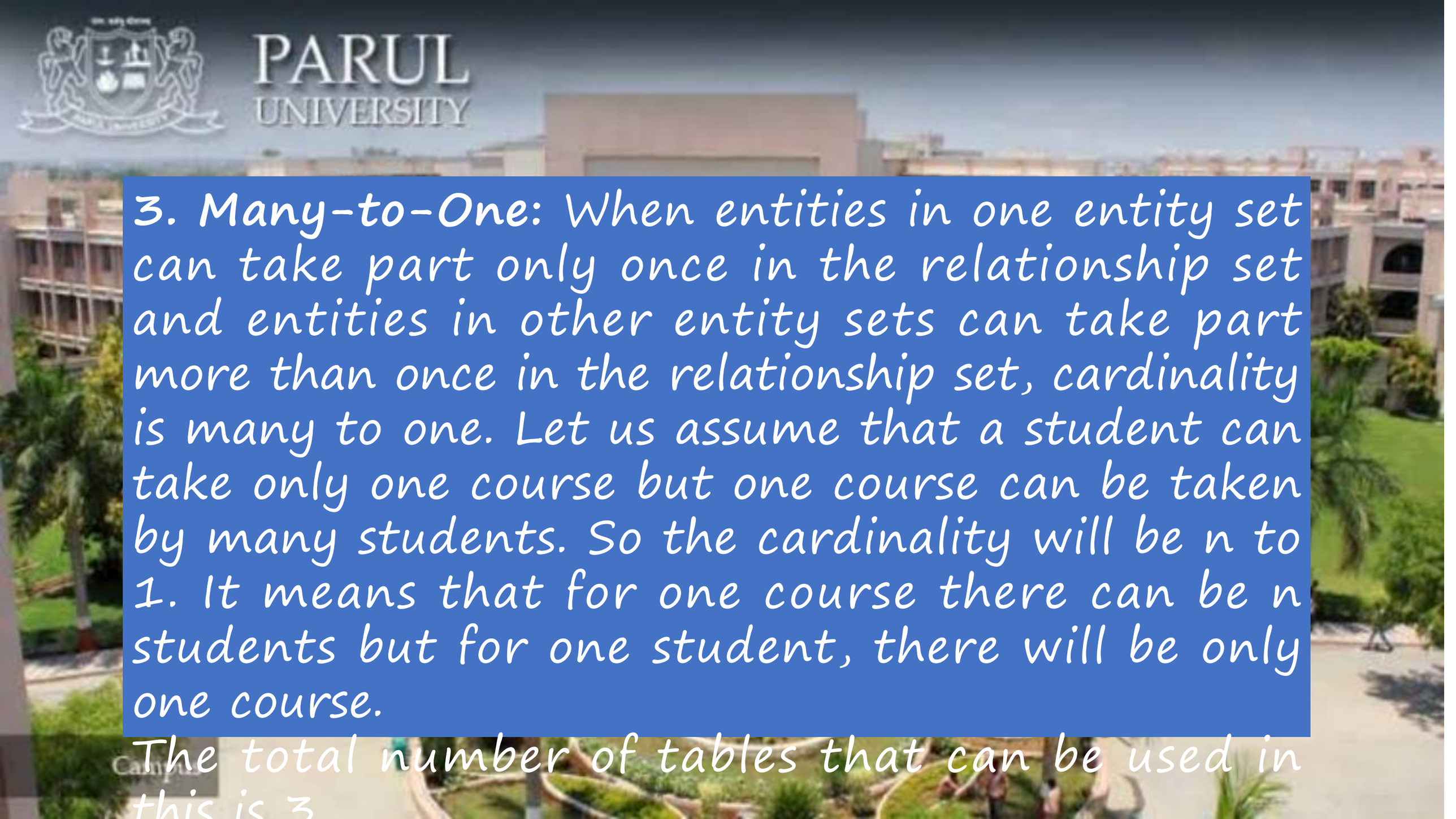



Surgeon Department
1
Has
M
Doctors


One-to-Many relationship
COLLEGE
has
DEPARTMENT

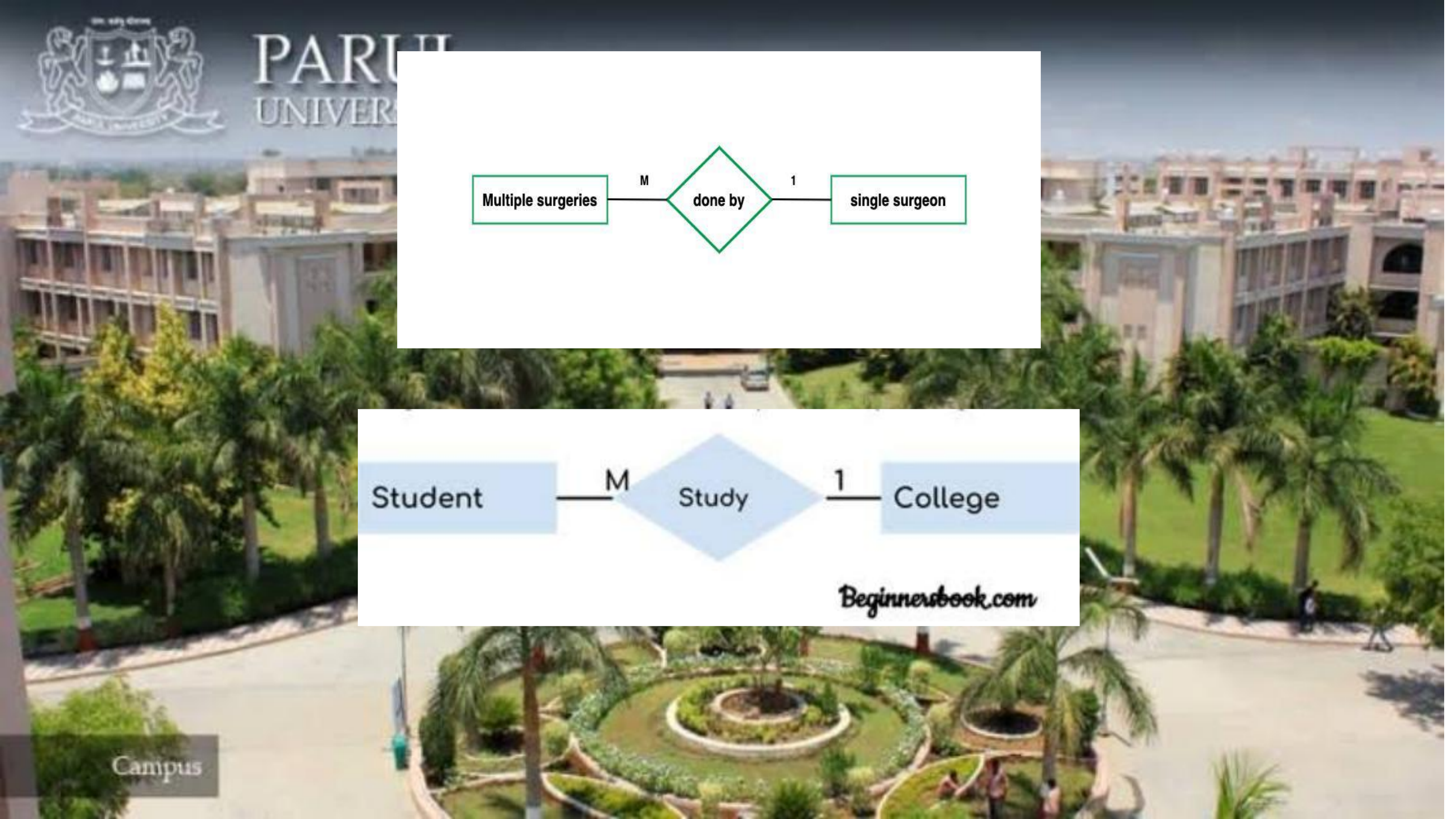

Customer
1
Places
*
Orders

Campus

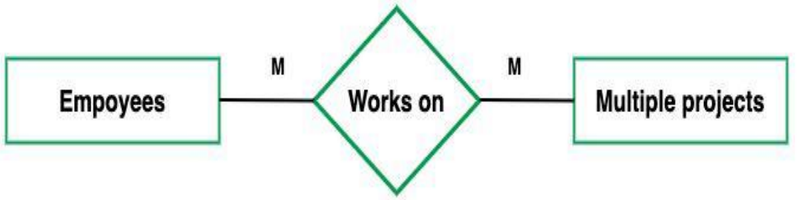


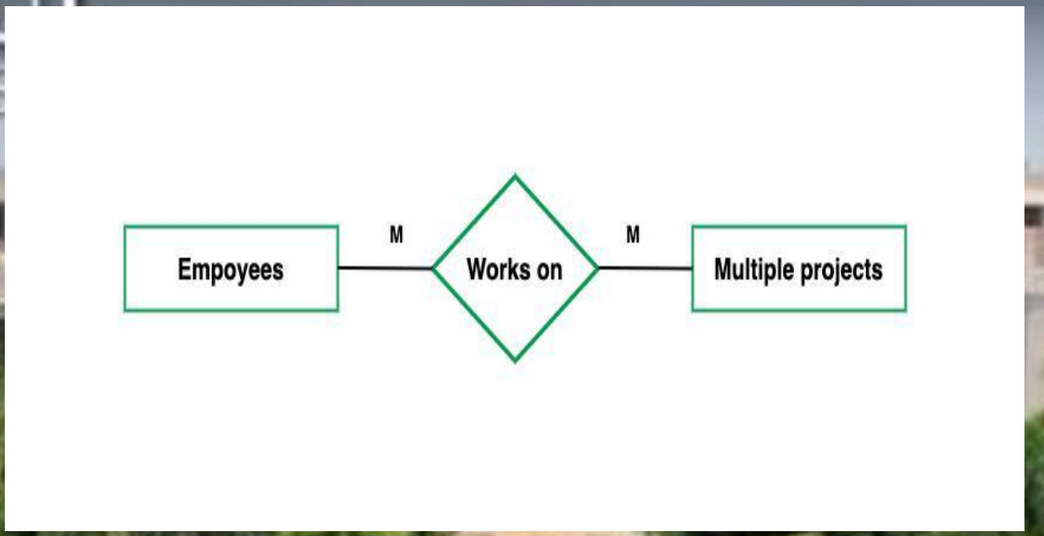
3. Many-to-One: When entities in one entity set can take part only once in the relationship set and entities in other entity sets can take part more than once in the relationship set, cardinality is many to one. Let us assume that a student can take only one course but one course can be taken by many students. So the cardinality will be n to 1. It means that for one course there can be n students but for one student, there will be only one course.

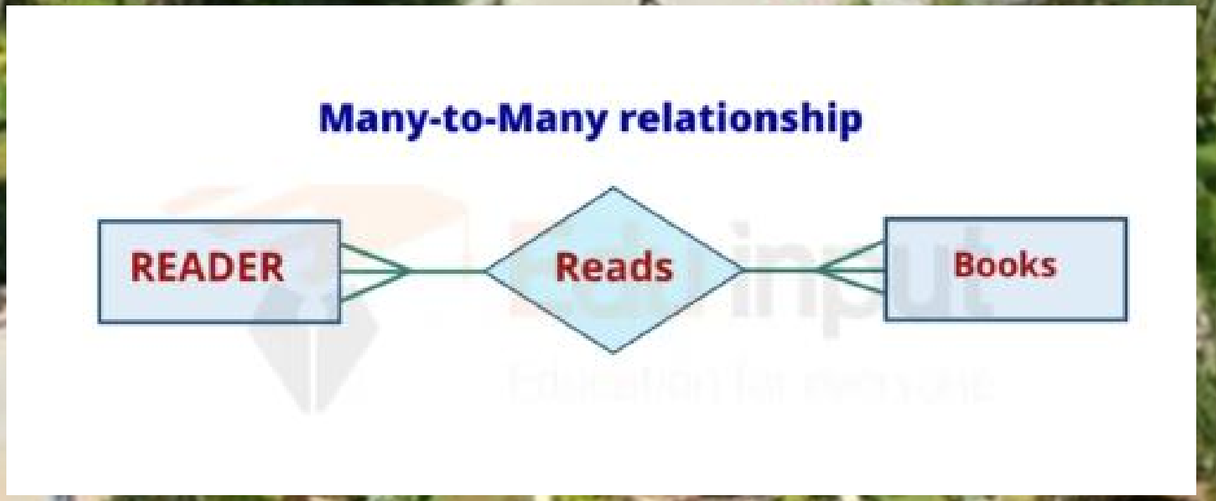
The total number of tables that can be used in this is 3.


PARU
UNIVER
Multiple surgeries
M
done by
1
single surgeon
Student
M
Study
1
College
Beginnersbook.com
Campus

4. Many-to-Many: When entities in all entity sets can take part more than once in the relationship cardinality is many to many. Let us assume that a student can take more than one course and one course can be taken by many students. So the relationship will be many to many.
the total number of tables that can be used in this is 3.




Empoyees
M
Works on
M
Multiple projects

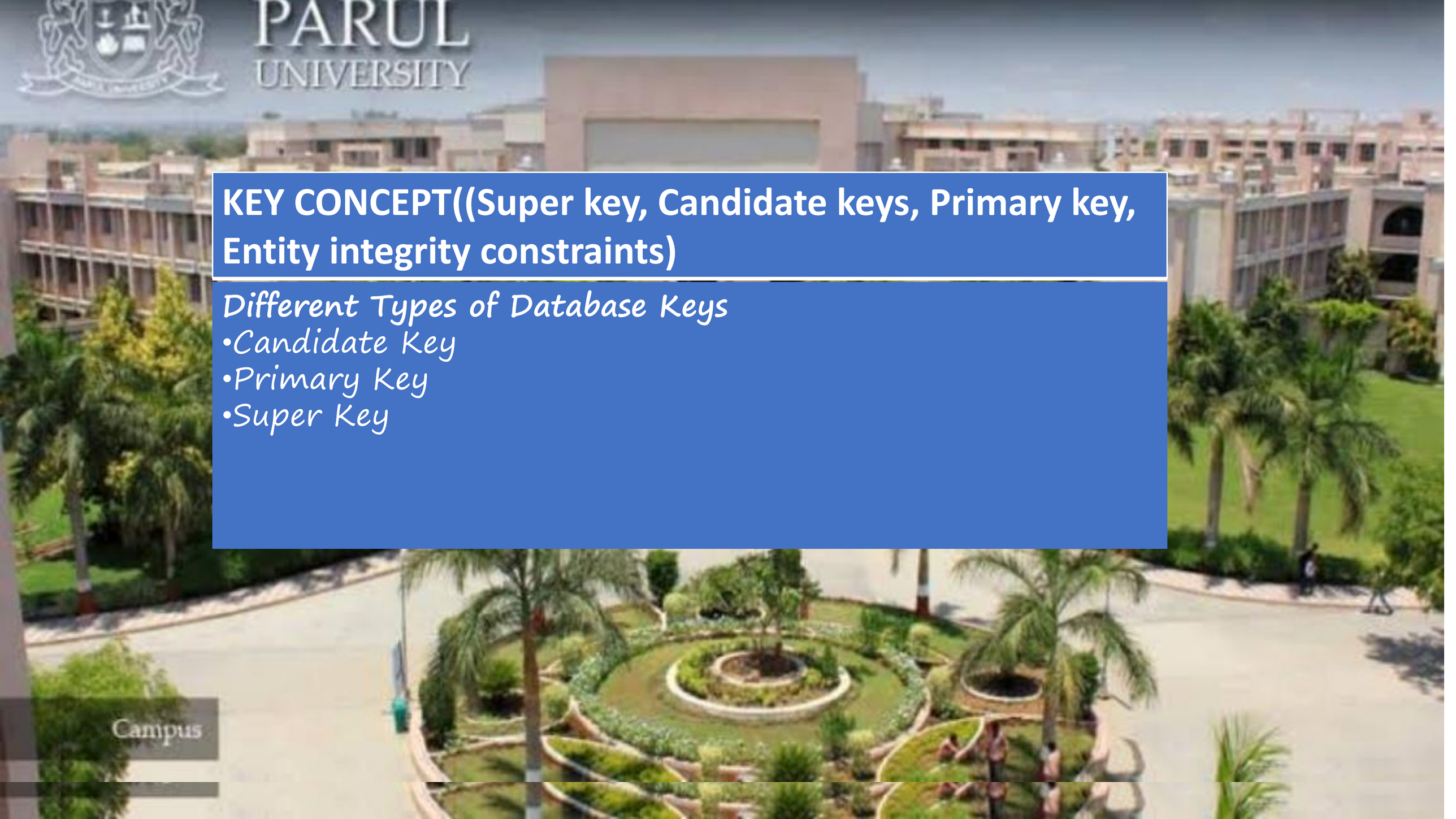

Many-to-Many relationship
READER
Reads
Books

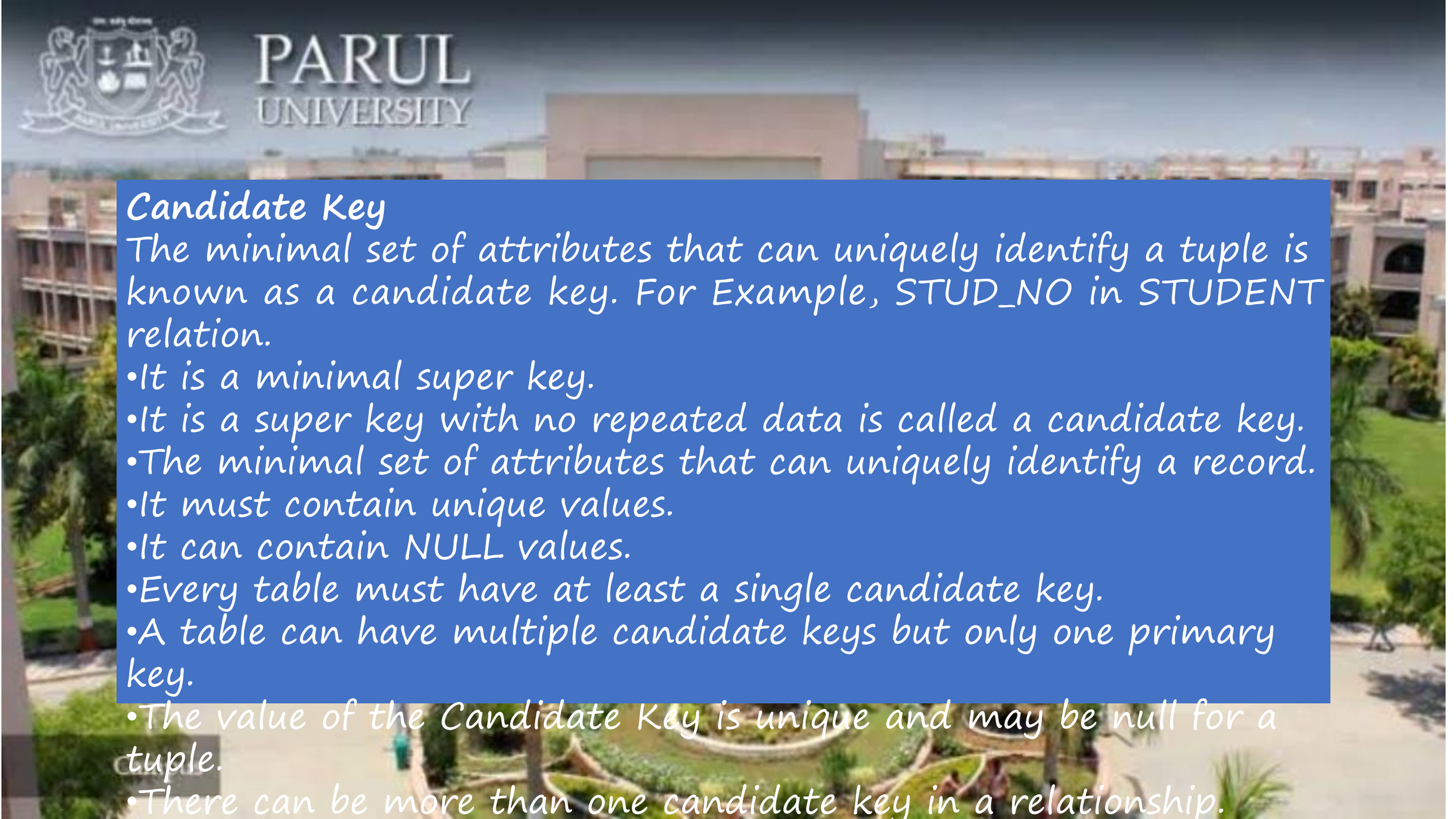
PARUL UNIVERSITY

KEY CONCEPT((Super key, Candidate keys, Primary key, Entity integrity constraints)

Different Types of Database Keys

- *Candidate Key*
- *Primary Key*
- *Super Key*





Candidate Key

The minimal set of attributes that can uniquely identify a tuple is known as a candidate key. For Example, STUD_NO in STUDENT relation.

- It is a minimal super key.
- It is a super key with no repeated data is called a candidate key.
- The minimal set of attributes that can uniquely identify a record.
- It must contain unique values.
- It can contain NULL values.
- Every table must have at least a single candidate key.
- A table can have multiple candidate keys but only one primary key.
- The value of the Candidate Key is unique and may be null for a tuple.
- There can be more than one candidate key in a relationship.

Example:

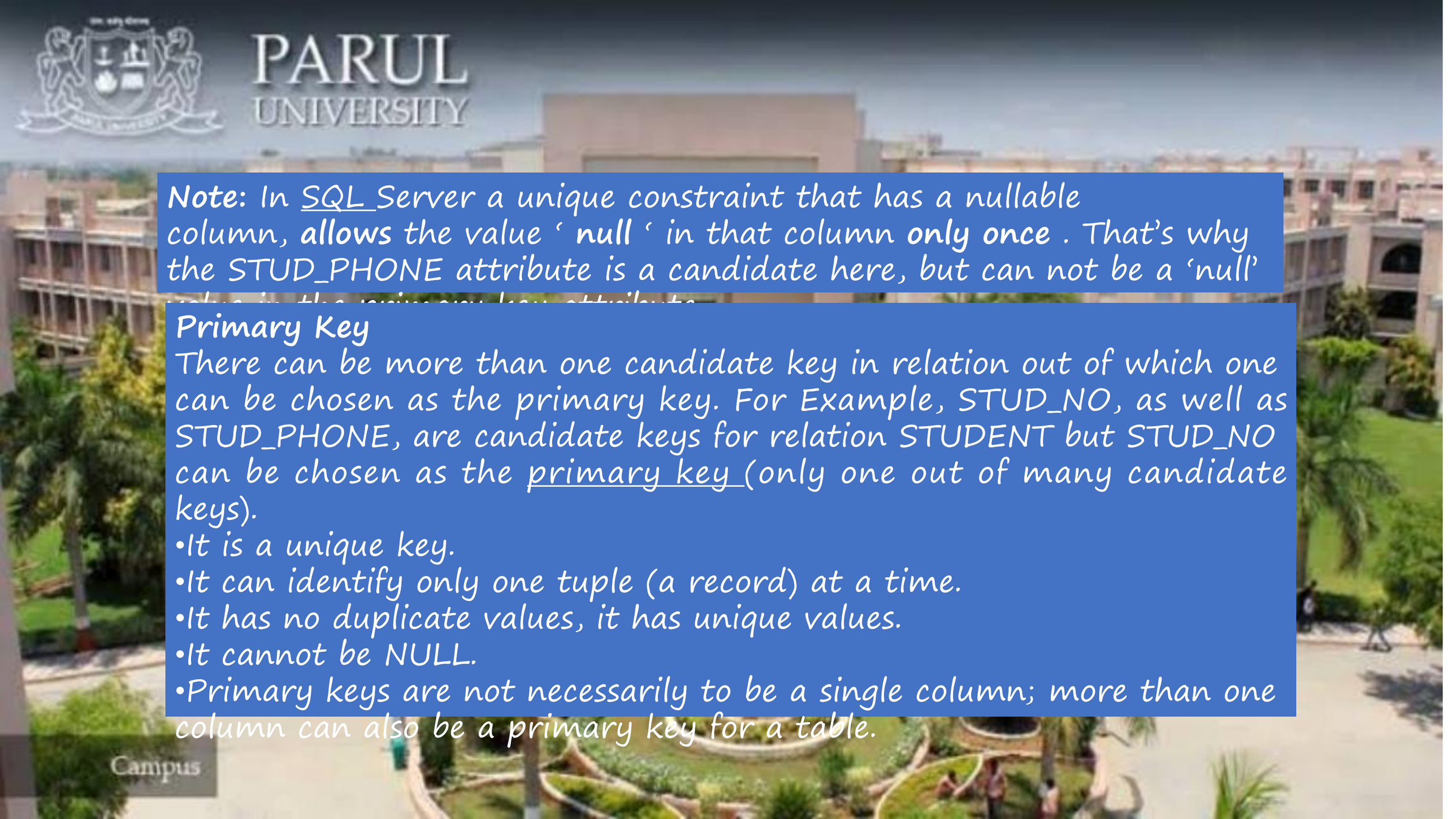
STUD_NO is the candidate key for relation STUDENT.

Table STUDENT

STUD_NO	SNAME	ADDRESS	PHONE
1	Shyam	Delhi	123456789
2	Rakesh	Kolkata	223365796
3	Suraj	Delhi	175468965

Example:

{STUD_NO, COURSE_NO} is a composite candidate key for relation STUDENT_COURSE.

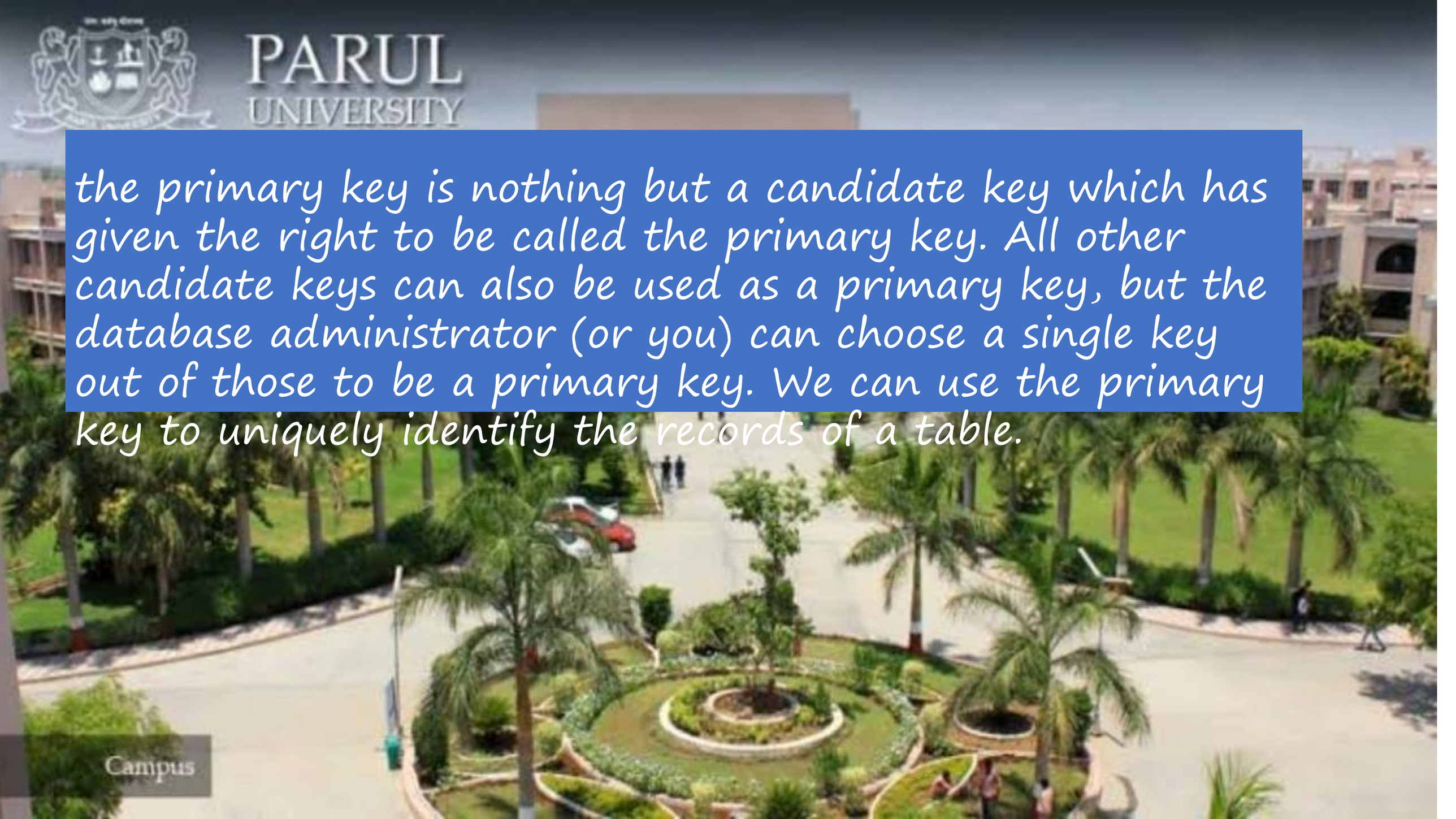


Note: In SQL Server a unique constraint that has a nullable column, **allows** the value ' **null** ' in that column **only once** . That's why the STUD_PHONE attribute is a candidate here, but can not be a 'null' value in the primary key attribute.

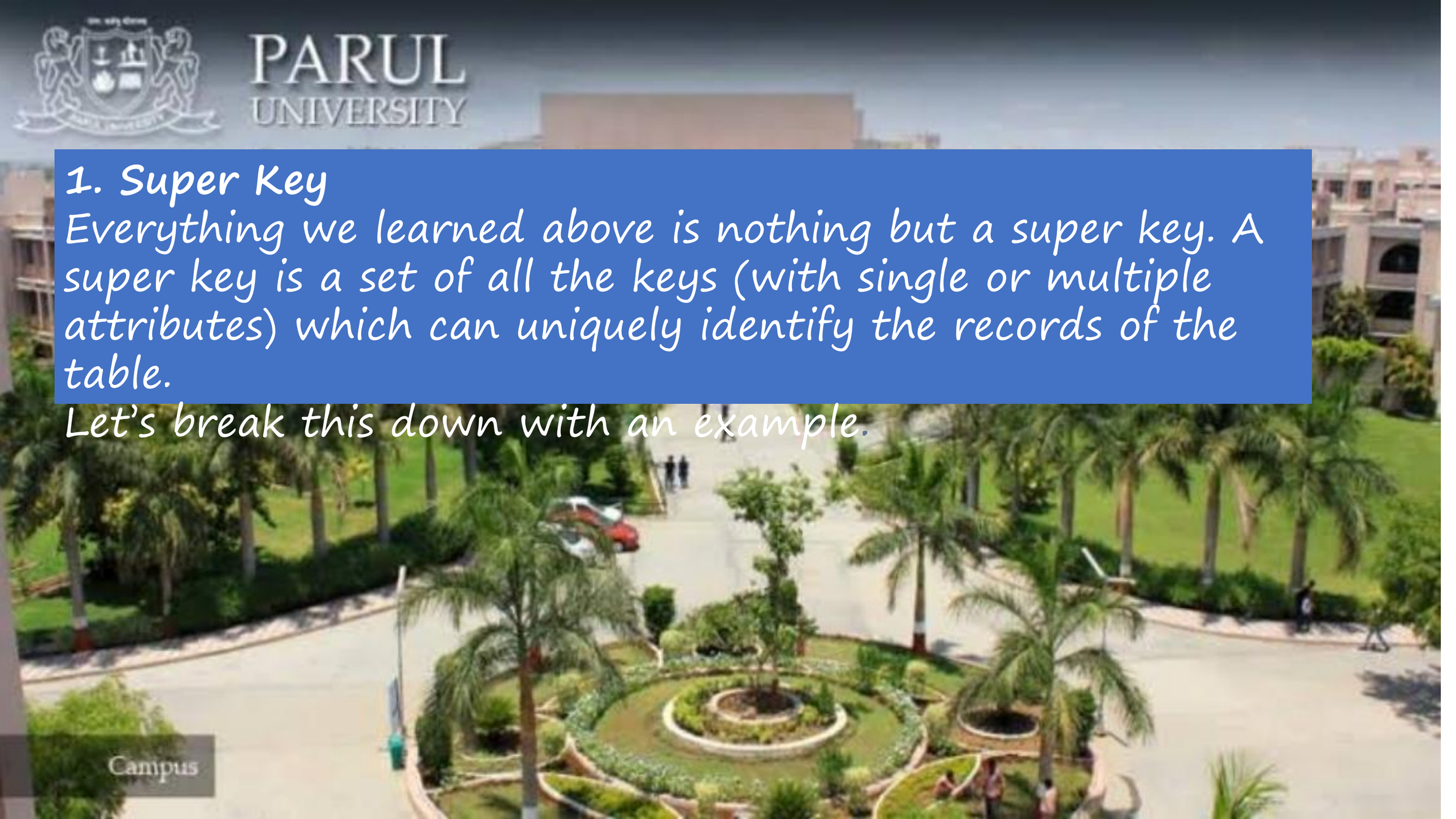
Primary Key

There can be more than one candidate key in relation out of which one can be chosen as the primary key. For Example, STUD_NO, as well as STUD_PHONE, are candidate keys for relation STUDENT but STUD_NO can be chosen as the primary key (only one out of many candidate keys).

- It is a unique key.
- It can identify only one tuple (a record) at a time.
- It has no duplicate values, it has unique values.
- It cannot be NULL.
- Primary keys are not necessarily to be a single column; more than one column can also be a primary key for a table.



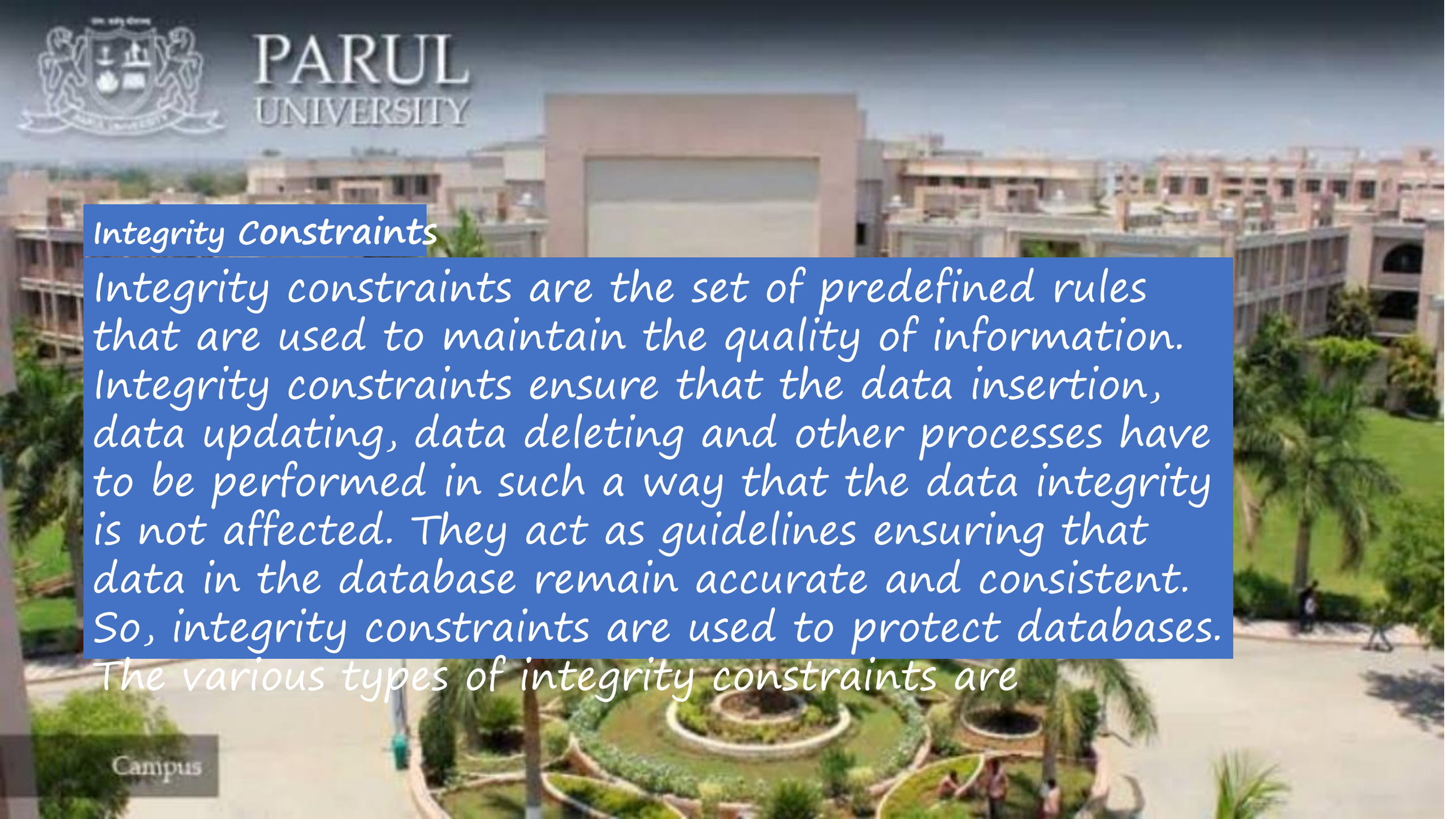
the primary key is nothing but a candidate key which has given the right to be called the primary key. All other candidate keys can also be used as a primary key, but the database administrator (or you) can choose a single key out of those to be a primary key. We can use the primary key to uniquely identify the records of a table.



1. Super Key

Everything we learned above is nothing but a super key. A super key is a set of all the keys (with single or multiple attributes) which can uniquely identify the records of the table.

Let's break this down with an example.



Integrity Constraints

Integrity constraints are the set of predefined rules that are used to maintain the quality of information. Integrity constraints ensure that the data insertion, data updating, data deleting and other processes have to be performed in such a way that the data integrity is not affected. They act as guidelines ensuring that data in the database remain accurate and consistent. So, integrity constraints are used to protect databases.

The various types of integrity constraints are

PARUL UNIVERSITY

Entity integrity constraints

Entity Integrity Constraints

Entity integrity constraints state that primary key can never contain null value because primary key is used to determine individual rows in a relation uniquely, if primary key contains null value then we cannot identify those rows. A table can contain null

Example:

It is not allowed because it is containing primary key as NULL value.

Student_id	Name	Semester	Age
21CSE101	Ramesh	5th	20
21CSE102	Kamlesh	5th	21
21CSE103	Aakash	5th	22
	Mukesh	5th	20

Campus

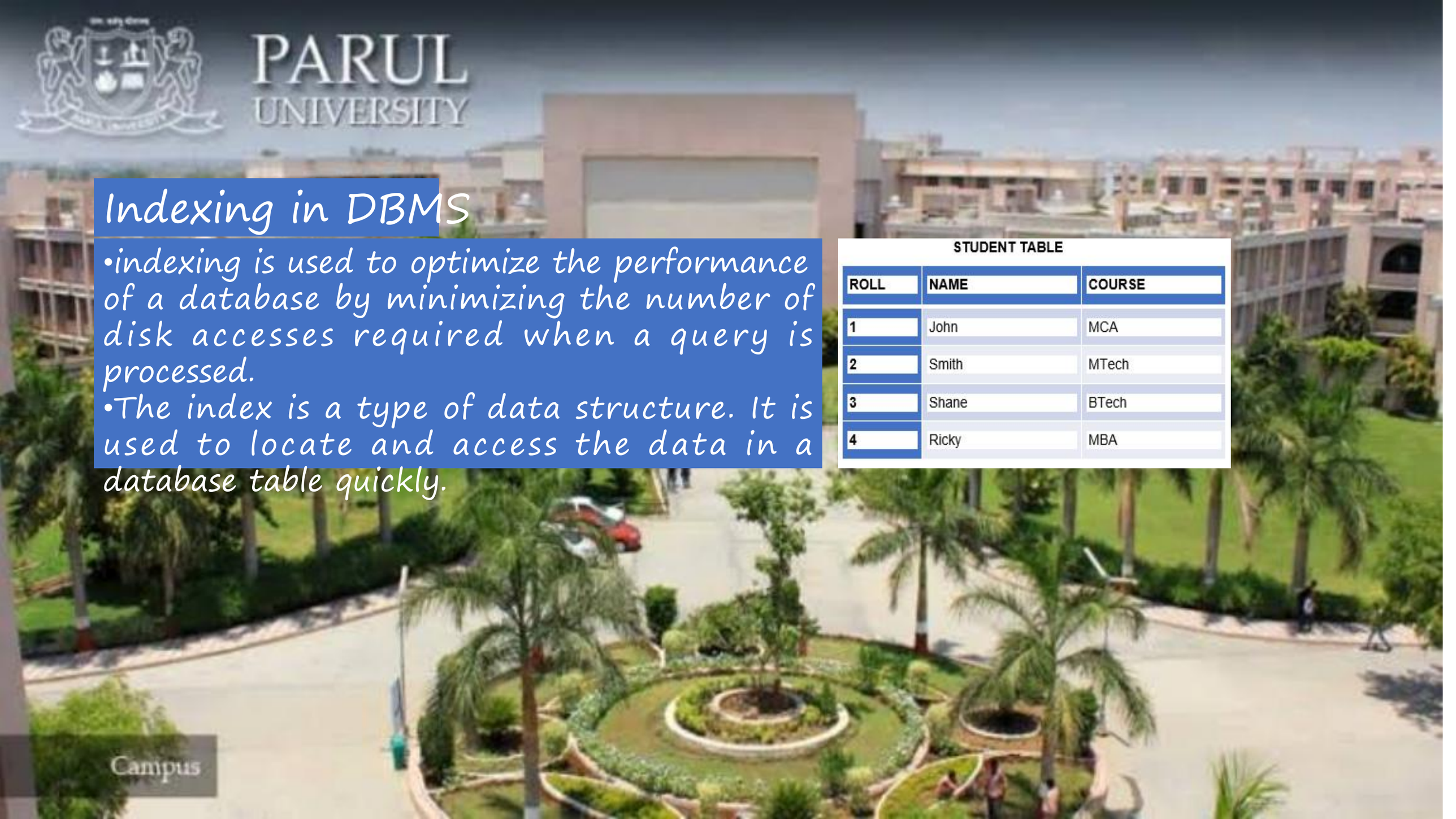
Referential Integrity constraints in DBMS

A referential integrity constraint is also known as **foreign key constraint**. A foreign key is a key whose values are derived from the Primary key of another table.

The table from which the values are derived is known as **Master or Referenced** Table and the Table in which values are inserted accordingly is known as **Child or Referencing** Table, In other words, we can say that the table containing the **foreign key** is called the **child table**, and the table containing the **Primary key/candidate key** is called the **referenced or parent table**. When we talk about the database relational model, the candidate key can be defined as a set of attribute which can have zero or more attributes.

The syntax of the Master Table or Referenced table is:

```
1.CREATE TABLE Student (Roll int PRIMARY KEY, Name varchar(25) , Course varchar(10) );
```



Indexing in DBMS

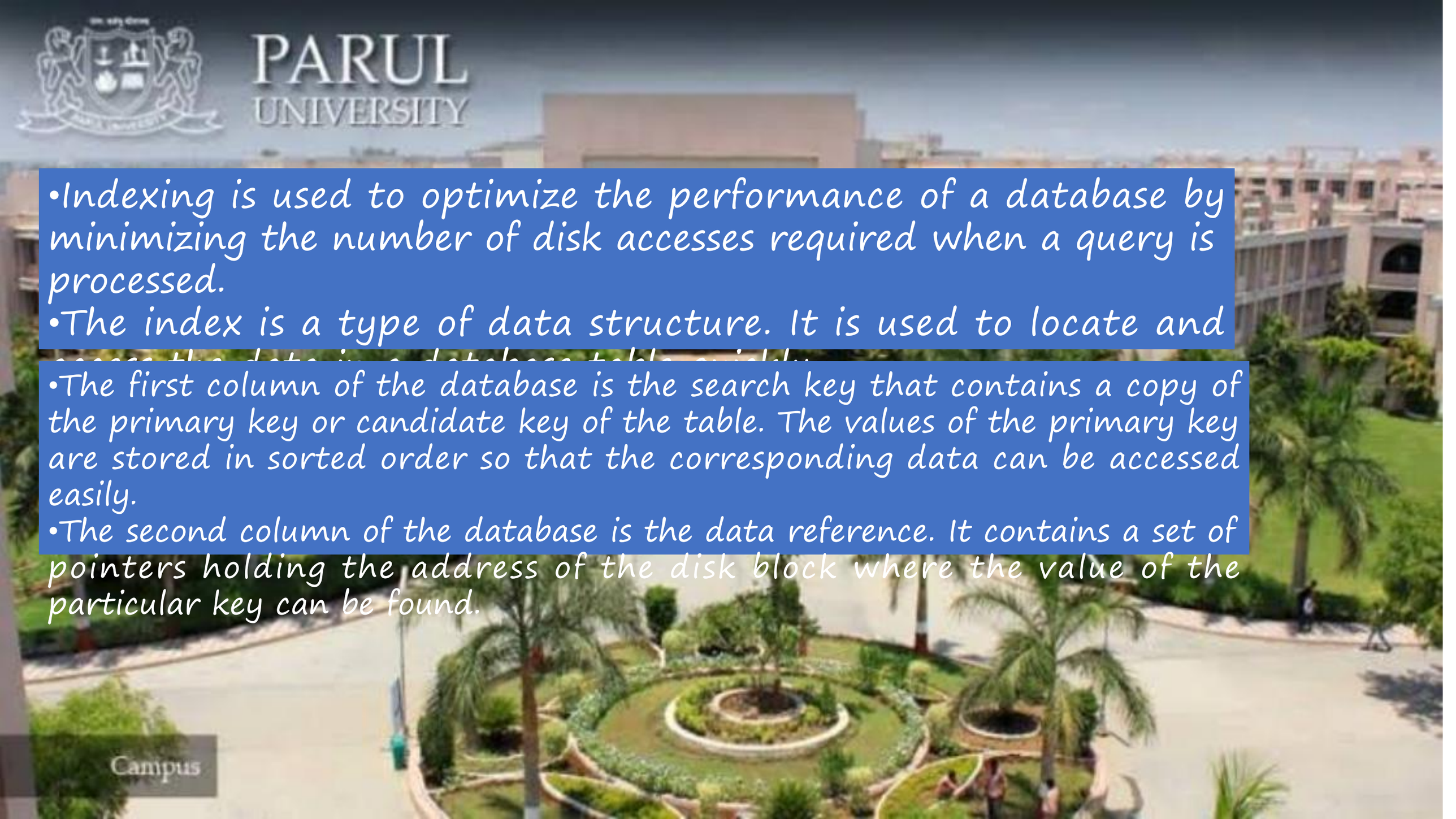
- indexing is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed.
- The index is a type of data structure. It is used to locate and access the data in a database table quickly.

STUDENT TABLE

ROLL	NAME	COURSE
1	John	MCA
2	Smith	MTech
3	Shane	BTech
4	Ricky	MBA


PARUL
UNIVERSITY
Campus


PARUL
UNIVERSITY
Campus



- Indexing is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed.
- The index is a type of data structure. It is used to locate and access the data in a database table quickly.
- The first column of the database is the search key that contains a copy of the primary key or candidate key of the table. The values of the primary key are stored in sorted order so that the corresponding data can be accessed easily.
- The second column of the database is the data reference. It contains a set of pointers holding the address of the disk block where the value of the particular key can be found.

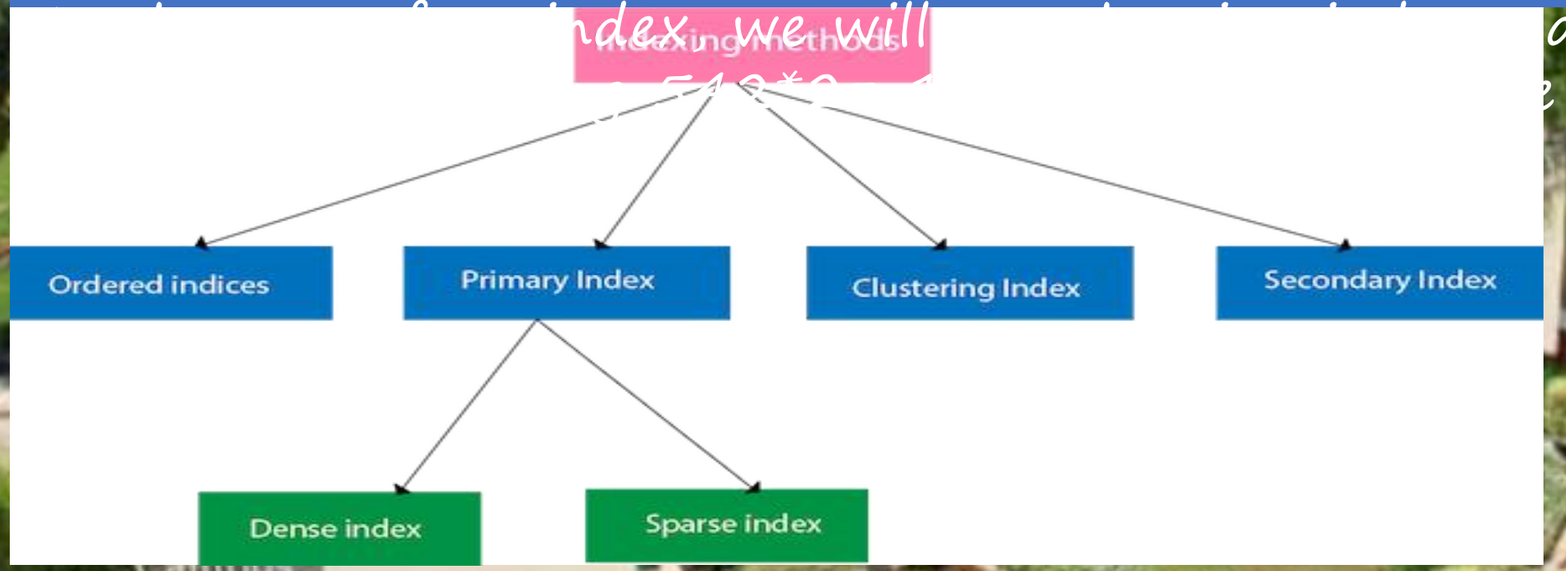
PARUL UNIVERSITY

Ordered indices

The indices are usually sorted to make searching faster. The indices which are sorted are known as ordered indices.

Example: Suppose we have an employee table with thousands of record and each of which is 10 bytes long. If their IDs start with 1, 2, 3....and so on and we have to search student with ID-543.

•In the case of a database with no index, we have to search the disk block from starting till it reaches 543. The DBMS will read the record after reading 543*10=5430 bytes.



ndex, we will
and the DBMS will read the
very less compared to the

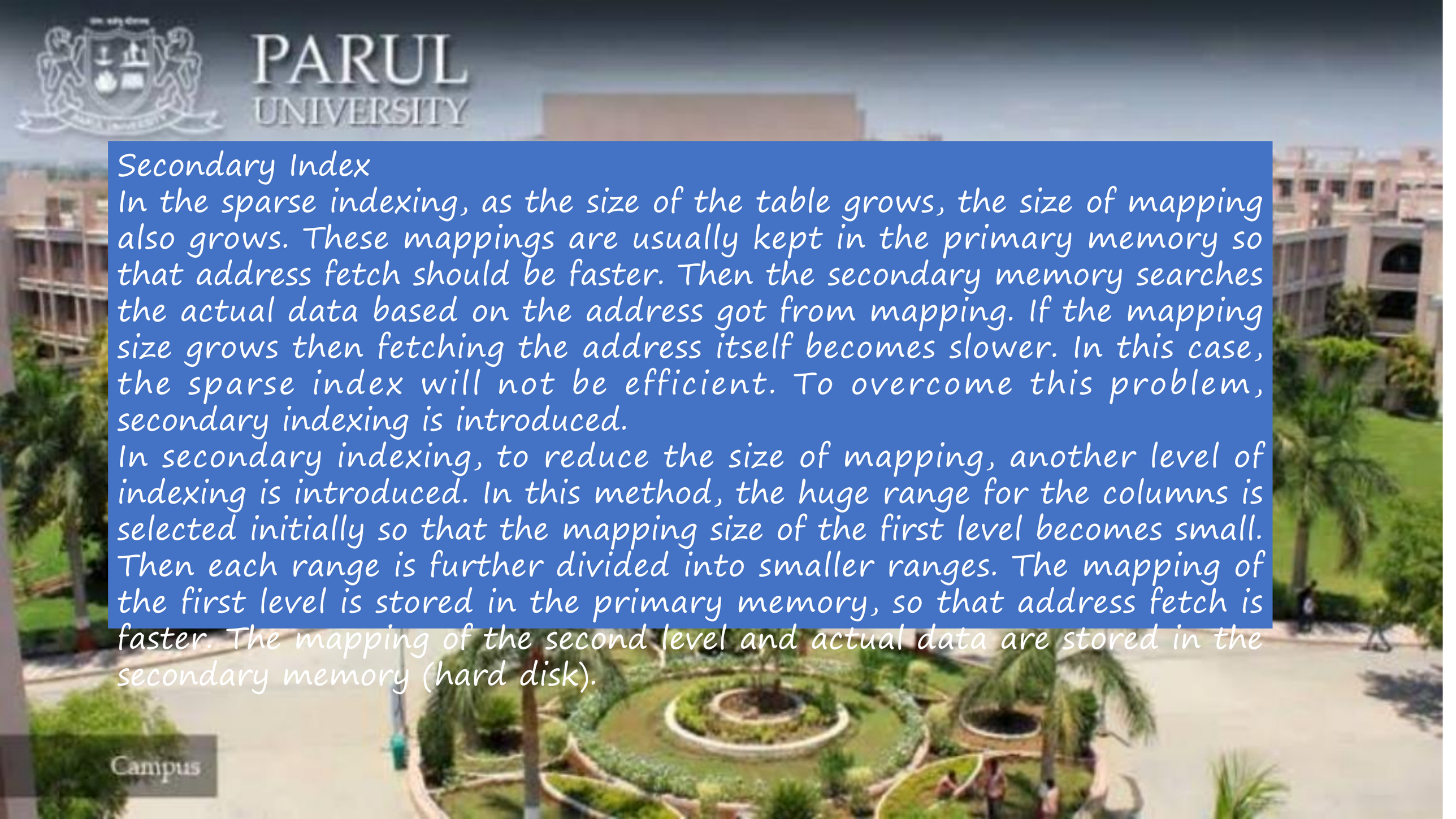
Primary Index

- If the index is created on the basis of the primary key of the table, then it is known as primary indexing. These primary keys are unique to each record and contain 1:1 relation between the records.
- As primary keys are stored in sorted order, the performance of the searching operation is quite efficient.
- The primary index can be classified into two types: Dense index and Sparse index.

Clustering Index

- A clustered index can be defined as an ordered data file. Sometimes the index is created on non-primary key columns which may not be unique for each record.
- In this case, to identify the record faster, we will group two or more columns to get the unique value and create index out of them. This method is called a clustering index.
- The records which have similar characteristics are grouped, and indexes are created for these group.

Example: suppose a company contains several employees in each department. Suppose we use a clustering index, where all employees which belong to the same Dept_ID are considered within a single cluster, and index pointers point to the cluster as a whole. Here Dept_Id is a non-unique key.



Secondary Index

In the sparse indexing, as the size of the table grows, the size of mapping also grows. These mappings are usually kept in the primary memory so that address fetch should be faster. Then the secondary memory searches the actual data based on the address got from mapping. If the mapping size grows then fetching the address itself becomes slower. In this case, the sparse index will not be efficient. To overcome this problem, secondary indexing is introduced.

In secondary indexing, to reduce the size of mapping, another level of indexing is introduced. In this method, the huge range for the columns is selected initially so that the mapping size of the first level becomes small. Then each range is further divided into smaller ranges. The mapping of the first level is stored in the primary memory, so that address fetch is faster. The mapping of the second level and actual data are stored in the secondary memory (hard disk).